Tree-Structured Networks Based on Polarity Shift and LSTM for Sentence Classification (Postprint)

Authors: Wang Ran, Jin Zhong

Date: 2018-05-20T00:00:00+00:00

Abstract

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of storing sequential information over long periods, which have been widely applied in language modeling, speech recognition, machine translation, and other domains. This paper first investigates how previous work extended the memory modules in LSTM to syntactic trees to obtain LSTM tree-structured network models, thereby capturing and storing deep semantic structural information of sentences. Then, addressing polarity shifts between words in sentences, polarity shift information is incorporated into the LSTM tree-structured network model, proposing a polarity-shift LSTM tree-structured network model that better captures sentiment information for sentence classification. Experimental results on the Stanford Sentiment Treebank dataset demonstrate that the proposed polarity-shift LSTM tree-structured network model achieves superior sentence classification performance compared to LSTM, recursive neural networks, and other models.

Full Text

Tree-Structured Networks Based on Polarity Shifting and LSTM for Sentence Classification

Wang Ran, Jin Zhong
(School of Computer Science & Engineering, Nanjing University of Science & Technology, Nanjing 210018, China)

Abstract: Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) that excels at preserving sequential information over extended periods and has been widely applied in language modeling, speech recognition, machine translation, and other domains. This paper first examines how previous research extended LSTM memory modules to syntax trees to obtain Tree-Structured

LSTM networks capable of capturing and storing deep semantic structural information from sentences. Then, addressing polarity shifting between words in sentences, we propose a Polarity Tree-Structured LSTM model that incorporates polarity shifting information into the Tree-Structured LSTM architecture to better capture sentiment information for sentence classification. Experiments on the Stanford Sentiment Treebank demonstrate that our proposed Polarity Tree-Structured LSTM model outperforms LSTM, recursive neural networks, and other baseline models for sentence classification.

Keywords: neural networks; LSTM; tree-structured network; polarity shifting; sentence classification

0 Introduction

Deep learning has been effectively and extensively applied to speech recognition [1], machine translation [2], and image-to-text conversion [3]. In natural language processing, continuous exploration of neural networks by researchers has led to significant advances in various neural network models for language modeling and text sentiment classification. Sentence classification holds substantial research value and practical importance in natural language processing [4], attracting increasing interest from scholars both domestically and internationally.

When using neural networks for sentence classification, the first challenge is sentence modeling—representing sentences through trained vectors. Currently, three models are widely employed: bag-of-words models, sequence models, and tree-structured models, each with distinct characteristics. Bag-of-words models treat sentences as unordered collections of words, sometimes using the average of word vectors as sentence representations [5,6]. While simple, this approach completely ignores word order and syntax, making it difficult to capture meaningful sentence information. Sequence models, by contrast, emphasize the arrangement of all words in a sentence [7,8], directly concatenating word vectors in sequential order to obtain sentence representations without considering structural information. Tree-structured models construct recursive networks based on given syntax trees to ultimately obtain representations of phrases and sentences within them [9,10]. Tree structures have been used to analyze natural image scenes and to transform word vectors into phrase vectors for sentiment polarity classification [11]. In 2013, Socher et al. applied recursive neural network models to sentiment classification on movie review data, demonstrating superior performance compared to traditional models.

However, bag-of-words models cannot fully grasp the complete semantics of natural language sentences and fail to distinguish semantic differences caused by word order or syntactic structure. Sequence and tree-structured models prove more effective, with tree-structured models being optimal due to their stronger correlation with syntactic structure. Recurrent Neural Networks (RNNs) provide an effective solution for sequential data, theoretically capable of processing

sequences of arbitrary length. However, training RNNs encounters the gradient vanishing problem [12]. LSTM networks, an RNN improvement, achieve significant breakthroughs in addressing gradient vanishing [13], enabling long-term preservation of sequential information. Nevertheless, LSTM networks still cannot obtain syntactic structure information from sentences. The Tree-Structured LSTM model, which applies LSTM memory modules within tree-based recursive neural network (RcNN) architectures, can capture both syntactic structure and semantic information while preserving node information throughout recursion.

1.1 Long Short-Term Memory Networks (LSTM)

Recurrent Neural Networks (RNNs) process arbitrary-length sequences through recurrent use of hidden state vectors within network blocks. At time step t, the hidden state vector h_t is obtained through a nonlinear transformation of the current input vector x_t and the previous hidden state vector h_{t-1}, as shown in the hyperbolic tangent function in Equation (1). However, RNNs suffer from gradient vectors that either vanish or explode during parameter training [15,16], particularly when processing long sequences, making it difficult to achieve good performance.

Long Short-Term Memory (LSTM) networks improve upon RNNs by utilizing a memory module that maintains multiple state vectors to preserve information at each time step. These state vectors include: input gate i_t, forget gate f_t, output gate o_t, memory cell state c_t, and hidden state h_t. The vector dimension d within the LSTM memory module is referred to as the internal vector dimension of the LSTM. The state vector transformations in an LSTM unit are defined as follows:

In Equations (2)~(7), x_t represents the input at time t, and the Sigmoid function constrains state vector values between 0 and 1. By employing various gating mechanisms to modify each vector's state, the model can handle problems across different time scales.

1.2 Tree-Structured Networks Based on LSTM

Words in sentences exhibit not only sequential relationships but also structural dependencies. Basic LSTM networks are limited by sequential ordering and cannot represent sentence structure information. Tree-structured networks can leverage syntactic trees constructed from sentences to build recursive networks that capture structural information, though they risk information loss. In 2015, Tai et al. combined the advantages of both approaches by extending LSTM memory modules to tree-structured networks, proposing two types of Tree-Structured LSTMs [17]: Dependency Tree LSTM (DTree-LSTM) and Constituency Tree LSTM (CTree-LSTM). These models have recently demonstrated excellent performance in sentence representation [18] and sentence analysis [19].

In Tree-Structured LSTM, a node' s LSTM memory module state vectors are all related to its child nodes. Node j contains state vectors including input gate i_j,

output gate o_j, memory cell state c_j, and hidden state h_j, where x_j represents the word vector in the sentence. Each child node is assigned a forget gate f_{jk} (where k is the child node index) to emphasize preservation of important semantic information.

1.2.1 LSTM-Enhanced Dependency Tree Networks

French linguist L. Tesniere first proposed dependency syntax. As shown in Figure 2(a), a dependency tree describes dependency relationships between nodes and their children, demonstrating syntactic and semantic collocations between words. Dependency trees have two key characteristics: child nodes are unordered, and the number of child nodes is variable. Based on these features, the state vector transformations for Dependency Tree LSTM (DTree-LSTM) are defined as follows:

In Equations (8)~(14), $C(j)$ denotes the set of child nodes of node j, and parameter matrices U and W transform the node' s x_j and h_k into internal vectors within the memory module. The DTree-LSTM network structure is illustrated in Figure 3(a).

1.2.2 LSTM-Enhanced Constituency Tree Networks

Constituency parsing originates from traditional sentence diagramming, which segments sentences into phrases (verb phrases, noun phrases, etc.) as shown in Figure 2(b). In these trees, non-leaf nodes represent relationships between child nodes, while leaf nodes represent words in the sentence. Constituency trees have two characteristics: they are binary trees with ordered child nodes, and only leaf nodes receive word inputs. Based on these features, the state vector transformations for Constituency Tree LSTM (CTree-LSTM) are defined as follows:

In Equations (15)~(20), N represents the number of child nodes, which is either 2 or 0 in constituency trees. This tree network assigns different parameter matrices to child nodes' hidden states, allowing different emphasis for phrase composition. Parameter matrices for nodes with strong sentiment should be adjusted to larger values during training. The CTree-LSTM network is shown in Figure 3(b).

2.1 Sentiment Polarity Shifting

In language understanding, word polarity shifting can influence the overall sentiment polarity of a sentence. Adverbs, negation words, and other linguistic elements can shift sentence polarity. As shown in Table 1, sentiment polarity shifting generally falls into four categories: three types of intra-sentence explicit shifting (intensification, negation, and contrast) and one type of inter-sentence polarity shifting [20]. In Tree-Structured LSTM, sentiment information at each node is ignored, prompting the proposal of PTree-LSTM networks that utilize

SoftMax to obtain node sentiment state vectors. These sentiment state vectors serve as polarity shifting features to better train the network. When computing node representations, the model incorporates not only semantic structure information preserved by child nodes in the original memory module but also child node sentiment information. Specifically, each node' s LSTM memory module includes an added sentiment state vector s_j and polarity shifting vector b_j.

2.2 Polarity-Shift-Enhanced Dependency Tree Networks

In dependency trees, child nodes are unordered, so child node sentiment state vectors need not be differentially weighted. The state vector transformations are defined in Equations (21)~(30).

In these equations, x_j represents word vectors in the sentence, and gate vectors i_j, f_{jk}, and o_j have values between 0 and 1. The forget gate f_{jk} controls which contents from child node memory modules will be discarded, the input gate i_j controls updates, and the output gate o_j controls output contents. In a dependency tree, if a node' s semantic information strongly expresses sentiment, the network training process will continuously adjust f_{jk} to make its values closer to 1 to preserve that node' s information, and vice versa.

The node's polarity shifting vector b_j is obtained by comprehensively considering child node sentiment state vectors s_k, representing polarity shifting information between words in the syntax tree. During network training, parameters are continuously adjusted to recursively transfer polarity shifting information from bottom to top, enabling more accurate acquisition of each node's sentiment label and the entire sentence' s sentiment information. The PDTree-LSTM network diagram is shown in Figure 4(a).

2.3 Polarity-Shift-Enhanced Constituency Tree Networks

In constituency trees, child nodes represent phrase combinations, and verb phrases express sentiment far more strongly than noun phrases. Therefore, child node sentiment state vectors cannot be treated equally as in dependency trees. The memory module assigns different parameter matrices to h_{jl} and h_{jr}, and network training continuously adjusts these parameters to emphasize preservation of verb phrase structural and sentiment information. Due to additional constraints on child nodes in constituency tree networks, these models contain far more parameters than dependency tree networks. Similarly, during training, the network continuously adjusts parameters to make node labels closer to ground truth labels. The PCTree-LSTM network diagram is shown in Figure 4(b).

2.4 Error Calculation for Sentence Classification with Tree-Structured Networks

This section describes how to apply the proposed models to sentence classification. Most nodes in tree-structured networks have ground truth labels, with each node representing the phrase rooted at that node. For any node j, the predicted label $\hat{y}_j$ can be obtained using Equations (39) and (40).

The sum of prediction errors between each node' s predicted label and ground truth label constitutes the entire network' s error. We employ the negative log-likelihood function as the error calculation function. For a tree structure with m nodes, each node' s error is calculated using Equation (41) (only for nodes with ground truth labels), where λ and θ are regularization hyperparameters.

3.1 Sentence Classification Experiments

To evaluate model effectiveness, we conducted experiments using the Stanford Sentiment Treebank [21] created from movie review data. The dataset contains two types: sentences for binary classification and for five-class classification (very negative, negative, neutral, positive, very positive). The dataset includes 11,855 sentences with an average length of 19 words and 215,154 phrases with provided phrase and sentence labels. For binary classification experiments, the dataset contains 6,920 training sentences, 872 validation sentences, and 1,821 test sentences (originally neutral sentences removed). For five-class experiments, it contains 8,544 training sentences, 1,101 validation sentences, and 2,210 test sentences. The dataset includes constituency trees for each sentence, and we used the Stanford semantic analysis toolkit to obtain dependency syntax trees.

In our experiments, we evaluated both PDTree-LSTM and PCTree-LSTM models on binary and five-class classification tasks. We used 300-dimensional GloVe vectors as word embeddings with a learning rate of 0.1 for word vectors during training. Models were trained using AdaGrad with a learning rate of 0.05 and batch size of 25. To prevent overfitting, we applied Dropout with a rate of 0.5 throughout model training. We found that the choice of internal vector dimension d in the memory module affects performance: smaller d preserves less information, while larger d leads to excessive parameters and overfitting. As shown in Figure 5, experiments revealed that $d = 100$ works best for binary classification and $d = 150$ for five-class classification.

Experimental results for LSTM, CTree-LSTM, PCTree-LSTM, DTree-LSTM, and PDTree-LSTM models on the validation set are shown in Figure 6. Our proposed PCTree-LSTM and PDTree-LSTM models demonstrate strong performance compared to baseline networks during training. Final test set comparisons with various network models are presented in Table 2, showing that both proposed models achieve excellent results in binary and five-class classification, with PCTree-LSTM performing best on this dataset, surpassing PDTree-LSTM. This performance gap may stem from differences in node counts between the

two tree structures. Constituency trees are phrase-based, containing 319,000 nodes with phrase labels, allowing them to capture more phrase information during training. Dependency trees are structure-based, with some nodes representing only word relationships whose subtrees do not form complete phrases, containing only 150,000 nodes with phrase labels. Additionally, constituency trees have far more parameters than dependency trees, and more parameters can be understood as preserving more information, which also affects classification performance.

4 Conclusion

This paper first introduced several highly effective existing networks, particularly the extension of LSTM memory modules to tree-structured networks, enabling effective acquisition of sentence semantic structure information and enriching sentence feature learning. We then proposed new network models by incorporating sentiment polarity shifting information into the LSTM memory modules of these tree-structured networks, further strengthening sentiment information learning. Neural network models offer significant advantages for sentence processing. Future work will focus on leveraging neural networks to more rapidly learn structural features of sentence text to more accurately obtain sentence sentiment information.

References

[1] Zhang Y, Chan W, Jaitly N. Very deep convolutional networks for end-to-end speech recognition [C]// Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. 2017: 4845-4849.

[2] Gulcehre C, Firat O, Xu K, et al. On integrating a language model into neural machine translation [J]. Computer Speech and Language, 2017, 45 (1): 137-148.

[3] Ma L, Lu Z, Li H. Learning to answer questions from image using convolutional neural network [C]// Proc of the 30th AAAI Conference on Artificial Intelligence. [S. l.] : AAAI Press, 2016: 3567-3573.

[4] 赵妍妍, 秦兵, 刘挺. 文本情感分析 [J]. 软件学报, 2010, 21 (8): 1834-1848.

[5] Landauer T K, Dumais S T. A solution to plato' s problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge [J]. Psychological Review, 1997, 104 (2): 211-240.

[6] Foltz P W, Kintsch W, Landauer T K. The measurement of textual coherence with latent semantic analysis [J]. Discourse Processes, 1998, 25 (2-3): 285-307.

[7] Elman J L. Finding structure in time [J]. Cognitive Science, 1990, 14 (2): 179-211.

[8] Mikolov T. Statistical Language models based on neural networks [J]. Presentation at Google, Mountain View, 2012.

[9] Goller C, Kuchler A. Learning task-dependent distributed representations by backpropagation through structure [C]// Proc of IEEE International Conference on Neural Networks, 1996: 347-352.

[10] Socher R, Lin C C, Manning C, et al. Parsing natural scenes and natural language with recursive neural networks [C]// Proc of the 28th International Conference on Machine Learning. 2011: 129-136.

[11] Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces [C]// Proc of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012: 1201-1211.

[12] Socher R, Pennington J, Huang E H, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2011: 151-161.

[13] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9 (8): 1735-1780.

[14] Jordan M I. Serial order: A parallel distributed processing approach [J]. Advances in Psychology, 1997, 121 (2): 471-495.

[15] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998, 6 (2): 107-116.

[16] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult [J]. IEEE Trans on Neural Networks, 1994, 5 (2): 157-166.

[17] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [C]// Proc of Association for Computational Linguistics. 2015.

[18] Rath T. Word and Relation Embedding for Sentence Representation [D]. Phoenix: Arizona State University, 2017.

[19] Goldberg Y. Neural Network Methods for Natural Language Processing [J]. Synthesis Lectures on Human Language Technologies, 2017, 10 (1): 1-309.

[20] 张小倩. 情感极性转移现象研究及应用 [D]. 苏州: 苏州大学, 2012.

[21] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2013: 1631-1642.

Note: Figure translations are in progress. See original paper for figures.