An Attention-Based Hybrid Neural Network for Document Modeling

Dengchao HE†, Hongjun ZHANG†(a), Wenning HAO†, Rui ZHANG†, Nonmembers, and Huan HAO††, Member

SUMMARY  The purpose of document modeling is to learn low-dimensional semantic representations of text accurately for Natural Language Processing tasks. In this paper, proposed is a novel attention-based hybrid neural network model, which would extract semantic features of text hierarchically. Concretely, our model adopts a bidirectional LSTM module with word-level attention to extract semantic information for each sentence in text and subsequently learns high level features via a dynamic convolution neural network module. Experimental results demonstrate that our proposed approach is effective and achieve better performance than conventional methods.

key words: document modeling, Bi-LSTM, attention-based model, dynamic CNN

1. Introduction

Document modeling is a fundamental task in the field of Natural Language Processing (NLP), which aims to learn low-dimensional semantic representations for documents. The significant challenge for modeling documents is to capture semantic features and to perform compositions over variable-length documents. Traditional approaches, such as the bag-of-words model consider documents as unordered word collections and rely on artificial designed features [1], which leads to fail to encode word orders and to extract semantic information [2].

In recent years, many deep neural network approaches are proposed to capture word order information and avoid designing labor intensive features. Most prevalent approaches based on neural networks fall into three categories: convolutional neural networks (CNNs) [3]–[6], recursive neural networks (RecursiveNNs) [7], [8] and recurrent neural networks (RecurrentNNs) [9], [10]. CNN based approaches adopt convolutional filters to extract local features over pretrained word embeddings. RecursiveNN based approaches build representations of phrases according to tree structures. RecurrentNN based approaches deliver semantic information in word sequence and are able to capture contextual information, which would be beneficial to model documents.

However, these approaches still remain several shortcomings. CNNs process word embeddings sequentially using sliding windows and treat document as a bag of n-grams, which would result in a loss of long dependency information and could not reflect semantic information completely. Then RecursiveNNs heavily rely on the performance of the parsing tree construction, which cannot be realized for many languages or noisy domains [11], [12] and would leads to error propagation or accumulation [13], [14]. And parsing tree is restricted to sentence-level modeling which is unsuitable to model documents. In addition, since RecursiveNNs would suffer the problem of vanishing gradient problem [15], fully labeling on internal nodes to supply additional supervisions is necessarily required. Furthermore, in RecurrentNNs, later semantic information are much more important than earlier words in a document. It would damage the performance of modeling documents, because significant information appears anywhere in a document rather than in the end.

In this paper, we propose a novel attention-based hybrid neural network model (ATT-HNN) for document modeling. Specifically, ATT-HNN first utilizes independent bidirectional long short-term memory network (Bi-LSTM) module networks to learn the representation of each sentence in documents with word-level attention. Then a Dynamic CNN layer is built on top of Bi-LSTM networks to extract semantic feature for a certain classification purpose. Our model is trained end-to-end with stochastic gradient descent, where the loss function is the cross-entropy error. We evaluate our ATT-HNN on several document classification tasks. Experimental results indicate that our proposed approach is effective and outperforms the conventional methods.

2. Methodology

Proposed ATT-HNN could learn text representations hierarchically from input raw word tokens without any other external supervision information, which consists of two modules. The bottom module is Bi-LSTM networks with word-level attention which would extract primary semantic features from each sentence in documents. The top is a dynamic CNN module which extracts high level semantic features from the Bi-LSTM networks’ outputs through one-dimensional wide convolution and dynamic k-Max pooling. Figure 1 gives the overall architecture of ATT-HNN.

Manuscript received December 4, 2016.
Manuscript revised February 11, 2017.
Manuscript publicized March 21, 2017.
†The authors are with the College of Command Information System, PLA University of Science and Technology, Nanjing, China.
††The author is with the College of Communication Engineering, PLA University of Science and Technology, Nanjing, China.
a) E-mail: hdciao1989@163.com
DOI: 10.1587/transinf.2016EDL8231
by [17]. Concretely, the LSTM unit contains one recurrently information that LSTM units should keep and memorize. In LSTM is to adopt memory blocks to decide the degree of decisions for the cell, i.e. an input gate \( i_t \), an output gate \( o_t \), and a forget gate \( f_t \). The LSTM unit at \( i \)-th word of the input depends on the previous state \( h_{t-1} \), the current input \( w_t \), and the memory cell \( c_{t-1} \). New vectors are calculated using the following equations.

\[
i_t = \sigma(W^{(i)} \cdot w_t + U^{(i)} \cdot h_{t-1} + b^{(i)}) \tag{1}
\]

\[
f_t = \sigma(W^{(f)} \cdot w_t + U^{(f)} \cdot h_{t-1} + b^{(f)}) \tag{2}
\]

\[
o_t = \sigma(W^{(o)} \cdot w_t + U^{(o)} \cdot h_{t-1} + b^{(o)}) \tag{3}
\]

\[
u_t = \tanh(W^{(u)} \cdot w_t + U^{(u)} \cdot h_{t-1} + b^{(u)}) \tag{4}
\]

\[
c_t = i_t \otimes u_t + f_t \otimes c_{t-1} \tag{5}
\]

where \( \sigma \) denotes the logistic function, \( \otimes \) denotes element-wise multiplication, \( W \) and \( U \) are weight matrices and \( b \) are bias vectors. The output of LSTM unit is the hidden state of recurrent networks, which is computed as follow:

\[
h_t = o_t \otimes \tanh(c_t) \tag{6}
\]

For document modeling, it is necessary to learn semantic information from past and future. However, conventional LSTM networks can only propagate and store history information, and cannot utilize and process information from future. Thus we employ Bi-LSTM networks [18] to extract complete semantic features. The core idea of Bi-LSTM is to present each sequence instance forwards and backwards to two separate semantic features. The overall architecture of Bi-LSTM is shown in Fig. 2.

![Fig. 2 The basic structure of a Bi-LSTM Network](image)

Then an average-pooling operator runs over all the hidden states \( h_t \) to obtain the sentence representation \( S_t \). It is observed that not all words in a sentence contribute equally to the semantic meaning for a certain classification task [19]. Therefore, we introduce word-level attention to achieve more beneficial semantic representations. Given the hidden states \( \{h_1, h_2, \ldots, h_n\} \) corresponding to the word in sentence \( S \), the sentence representation \( S \) is obtained as follow:

\[
S = \frac{1}{n} \sum_{i=1}^{n} \alpha_i h_i \tag{7}
\]

where \( \alpha_i \) are the word-level attention weights, which are defined as:

\[
\alpha_i = \frac{\exp(e(h_i))}{\sum_j \exp(e(h_j))} \tag{8}
\]

In Eq. (9), \( e(\cdot) \) is a measure function which reflects the relevance between each word and corresponding the purpose vector. Inspired by [20], \( e(\cdot) \) is defined as:

\[
e(h_i) = h_i \cdot v \cdot P \tag{9}
\]

where \( v \) is a weighted diagonal matrix and \( P \) is the representation vector of the specific purpose which would be learned in training phrase. Afterwards, we would acquire the semantic representations of each sentence.

### 2.2 Dynamic CNN Module

It is well known that the composition of words builds up the semantic meaning for sentences. Therefore, it would be inferred that the composition of sentences establishes the semantic meaning for documents. Thus, a dynamic CNN
module follows as the second module in ATT-HNN, which utilizes the dynamic \( k \)-Max pooling [3] to extract high level semantic features of documents.

Given a set of sentence representations \( \{S_1, S_2, \ldots, S_m\} \), dynamic CNN adopts one-dimensional wide convolution [3] to obtain a local feature sequence \( \mathbf{c} = [c_1, c_2, \ldots, c_{m+1}] \). Concretely, a local feature \( c_i \) would be acquired from a window of \( S_{i+l-1} \) by a filter with size \( l \) as follows:

\[
c_i = f(W \cdot S_{i+l-1} + b)
\]

(10)

It is notable that when calculating Eq. (10), out of range input value \( S_i \) where \( i < 1 \) or \( i > m \) are taken to be zero. Then the local feature sequence \( \mathbf{c} \) is fed to a pooling layer. ATT-HNN employs dynamic \( k \)-Max pooling to sample and select features. Given feature matrix \( \mathbf{c} \), dynamic \( k \)-Max pooling operator gives a dynamic value \( k \) with \( k < s + l - 1 \), and selects the \( k \) highest values of each row of \( \mathbf{c} \) to generate \( \mathbf{c}_{k\text{-max}} \):

\[
c_{k\text{-max}} = k - \max[\mathbf{c}] = \left[ \begin{array}{c} k - \max(c_1, :) \\ \vdots \\ k - \max(c_s, :) \end{array} \right]
\]

(11)

where \( d \) is the dimension of feature \( c_i \). Let \( L \) be the total number of dynamic CNN layers and \( l \) is the number of the current dynamic CNN layer; \( k_{top} \) is a fixed parameter for the topmost dynamic CNN layer, \( n \) is the length of input sequence, then we calculate \( k \) as follows:

\[
k = \max(\left\lfloor \frac{L - l}{L} \times n, k_{top} \right\rfloor)
\]

(12)

The \( k \)-Max pooling operator pools the \( k \) most prominent features and preserves the order of features. As the length of input sequence varies, the size of feature vector \( \mathbf{c}_{k\text{-max}} \) in \( \mathbf{c} \) is changing dynamically. And at the topmost layer, the final feature \( \mathbf{c}_{k\text{-max}} \) is fixed as \( d \times k_{top} \). Thus \( k \)-Max pooling is suitable for processing documents. Finally, a Full-connected layer and a softmax layer follow to perform the specific classification task.

3. Experiments

Extensive experiments have been carried out to verify that our proposed ATT-HNN model is effective for document modeling.

3.1 Experimental Settings

We perform the experiments using the following three datasets: Yelp 2013, Yelp 2014, Yelp 2015 from Yelp Dataset Challenge and IMDB built by [21]. All the datasets are separated into training, development and testing sets in the proportion of \( 8 : 1 : 1 \), and are preprocessed using following steps [22]: (1) removal of paragraphs that are not in English. (2) adopt the Stanford Tokenizer\(^\dagger\) to obtain raw word tokens. (3) substitution of non-western characters for a special character. The word embeddings used in experiments are 300-dimensional pre-trained GloVe embeddings\(^\ddagger\). Adadelta is adopted [23] to update parameters. Other hyper-parameters are tuned through 10-fold validation on the training sets, which are listed in Table 1.

\(^\dagger\)http://nlp.stanford.edu/software/tokenizer.shtml

\(^\ddagger\)http://nlp.stanford.edu/projects/glove/

Table 1 Hyper-parameters adopted in the experiments

| Word Embeddings | Filter Size | \( k_{top} \) | Dropout Rate | Batch Size | Adam\(\epsilon \) Parameter |
|-----------------|-------------|--------------|--------------|------------|--------------------------|
| DesignedFeatures | AvgWordEmbeddings | RNTN+RNN | Paragraph Vector | CNN | RCNN | HNN | ATT-HNN |
| AvgWordEmbeddings | 35.3 | 54.0 | 55.4 | 30.4 | AvgWordEmbeddings | 58.7 | 61.8 | 62.4 | 40.2 | AvgWordEmbeddings | 58.4 | 59.7 | 60.2 | 37.2 | AvgWordEmbeddings | 57.7 | 59.4 | 60.5 | 34.1 | AvgWordEmbeddings | 59.7 | 61.0 | 61.5 | 37.6 | AvgWordEmbeddings | 61.4 | 63.1 | 63.9 | 41.3 |
| HNN | 59.4 | 61.3 | 61.7 | 42.6 | ATT-HNN | 62.9 | 63.2 | 65.2 | 41.2 |

Table 2 The accuracy of document classification for the four datasets (%). The best method in each setting is in bold.

| Dataset | 2013 | 2014 | 2015 | IMDB |
|---------|------|------|------|------|
| Yelp    | 61.8 | 62.4 | 55.4 | 37.2 |
| Yelp    | 59.7 | 61.0 | 30.4 | 42.6 |
| Yelp    | 61.4 | 63.1 | 29.4 | 41.3 |
| IMDB    | 62.9 | 63.2 | 65.2 | 41.2 |

Table 2 demonstrates the results of ATT-HNN on four experimental datasets, indicating the effectiveness in comparison with other conventional methods. It is observed that ATT-HNN significantly achieves better performance than most conventional approaches.

Concretely, from experiment results we observed that designed features are effective for document modeling, but it could not outperform \( n \)-gram approaches. AvgWordEmbeddings is an approach without feature designing. However, it seems to be inappropriate in this scenario. When comparing conventional neural network approaches (RNTN+RNN, Paragraph Vector, CNN and RCNN) with above AvgWordEmbeddings, we can indicate that deep semantic compositionality is benefit for encoding documents. RNTN+RNN adopts Recursive Neural Tensor Network (RNTN) to obtain sentence embeddings and propagate them to a RecurrentNN to model documents. Paragraph Vector and CNN utilize convolutional operator to acquire document representations. Among all these methods, RCNN achieves a significantly performance. RCNN combines a bi-directional recurrent structure and max-pooling layer to utilize the advantage of both RecurrentNNs and CNNs, which would extract the semantic information across sentences completely.
When comparing our ATT-HNN model to other neural network approaches (RNTN+RNN, Paragraph Vector, CNN, RCNN and HNN), we can find that ATT-HNN obtains higher accuracy than others. It is illustrated that the word-level attention mechanism could obtain a considerable improvement. This is because word-level attention mechanism is adopted, which improves the weights of meaningful words in document modeling. And also relations between sentences are well captured by our ATT-HNN.

4. Conclusion

In this paper, we propose a novel neural network model for document modeling, ATT-HNN, which realizes automatically learning continuous document representation. Specifically, ATT-HNN utilizes independent Bi-LSTM networks with word-level attention to learn representation of each sentence in documents. Afterwards, a dynamic CNN layer is built on top of Bi-LSTM networks to extract semantic feature for a certain classification purpose. Experimental results indicate that our approach is effective and outperforms conventional methods.

Acknowledgments

This work has been supported by The Jiangsu Province Nature Science Foundation of China [Grant No. BK20150720].

References

[1] E. Bingham, A. Kabán, and M. Girolami, “Topic identification in dynamical text by complexity pursuit,” Neural Processing Letters, vol.17, no.1, pp.69–83, 2003.

[2] Y. Bengio, I. Goodfellow, and A. Courville, Deep learning, MIT Press, 2015.

[3] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” arXiv:1404.2188, 2014.

[4] Y. Kim, “Convolutional neural networks for sentence classification,” Proc. EMNLP, pp.1746–1751, 2014.

[5] W. Yin and H. Schütze, “Multichannel variable-size convolution for sentence classification,” Proc. CoNLL, pp.204–214, 2015.

[6] S. Zhou, Q. Chen, and X. Wang, “Convolutional deep networks for visual data classification,” Neural processing letters, vol.38, no.1, pp.17–27, 2013.

[7] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” Proc. EMNLP, pp.1631–1642, 2013.

[8] K. Tai, R. Socher, and C.D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” arXiv:1503.00075, 2015.

[9] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model,” Spoken Language Technology Workshop, IEEE, pp.234–239, 2012.

[10] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” Proc. ICML, pp.1188–1196, 2014.

[11] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H.D. Lii, “Deep unordered composition rivals syntactic methods for text classification,” Proc. ACL, pp.1681–1691, 2015.

[12] M. Ma, L. Huang, B. Xiang, and B. Zhou, “Dependency-based convolutional neural networks for sentence embedding,” Proc. ACL, pp.174–179, 2015.

[13] S. Wen and Z. Zeng, “Dynamics analysis of a class of memristor-based recurrent networks with time-varying delays in the presence of strong external stimuli,” Neural processing letters, vol.35, no.1, pp.47–59, 2012.

[14] D. Zeng, K. Liu, Y. Chen, et al., “Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks,” Proc. EMNLP, pp.17–21, 2015.

[15] C. Privault and L. Hérault, “Constraints satisfaction through recursive neural networks with mixed penalties: a case study,” Neural processing letters, vol.8, no.1, pp.15–26, 1998.

[16] B. Sjö, “Selective Recurrent Neural Network,” Neural processing letters, vol.38, no.1, pp.1–15, 2013.

[17] W. Zaremba and I. Sutskever, “Learning to execute,” arXiv:1410.4615, 2014.

[18] M. Schuster and K.K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol.45, no.11, pp.2673–2681, 1997.

[19] H. Chen, M. Sun, C. Tu, et al., “Neural Sentiment Classification with User and Product Attention,” Proc. EMNLP, 2016.

[20] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, “Neural relation extraction with selective attention over instances,” Proc. ACL, pp.2124–2133, 2016.

[21] Q. Diao, M. Qiu, C.-Y. Wu, A.J. Smola, J. Jiang, and C. Wang, “Jointly modeling aspects, ratings and sentiments for movie recommendation,” SIGKDD ACM, pp.193–202, 2014.

[22] A. Maas, R. Daly, P. Pham, D. Huang, et al., “Learning word vectors for sentiment analysis,” Proc. ACL-HLT, pp.142–150, 2011.

[23] M. Zeiler, “Adadelta: An adaptive learning rate method,” arXiv:1212.5701, 2012.

[24] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” Proc. ICML, pp.1188–1196, 2014.

[25] S. Lai, L. Xu, et al., “Recurrent Convolutional Neural Networks for Text Classification,” Proc. AAAI, pp.2267–2273, 2015.