SCIENCE CHINA
Information Sciences

• LETTER •

June 2022, Vol. 65 169103:1–169103:3
https://doi.org/10.1007/s11432-019-2775-9

Densely-connected neural networks for aspect term extraction

Chen CHEN[1*], Houfeng WANG[2*], Qingqing ZHU[1] & Junfei LIU[3]

[1]*School of Software and Microelectronics, Peking University, Beijing* 100871*, China;*
[2]*MOE Key Lab of Computational Linguistics, School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871*, China;*
[3]*National Engineering Research Center for Software Engineering* (*Peking University*)*, Beijing* 100871*, China*

Received 14 June 2019/Revised 7 September 2019/Accepted 14 November 2019/Published online 7 September 2021

Citation Chen C, Wang H F, Zhu Q Q, et al. Densely-connected neural networks for aspect term extraction. Sci China Inf Sci, 2022, 65(6): 169103, https://doi.org/10.1007/s11432-019-2775-9

Dear editor,

Aspect term extraction (ATE) is a sub-task of aspect-based sentiment analysis, which aims to extract opinionated aspect terms from user reviews. For example, in a laptop domain review: "Boot time is super fast", boot time is an aspect, and the sentiment towards it is positive, which can be inferred from super fast.

Existing approaches to solving the ATE task could be categorized as unsupervised, weakly supervised, and supervised. Research on supervised methods usually treats the task as a token-level sequence labeling problem and focuses on extracting features. Recently, automated feature learning by deep neural networks is preferred because it is difficult to obtain useful features manually [1–3]. Most models, however, only use the representations of the last layer as features for prediction. Peters et al. [4] showed that the representations vary with network depth in neural networks: the morphological information is encoded at the word embedding layer; the local syntax is captured at lower layers; the longer-range semantics are encoded at the upper layers. Thus, a traditional multi-layer neural model could lose the low-level features, while a single-layer neural model cannot obtain the high-level features.

Inspired by [5], we propose a densely-connected multi-layer neural network model for ATE that can combine the features from each layer. Specifically, the model contains three components (1) The double embedding mechanism combines general embeddings and domain embeddings to improve the quality of embeddings; (2) The multi-layer BiLSTM networks process the inputs in forward and backward directions to generate the token-level representations by recording the sequential information; (3) The self-attention mechanism conducts direct connections between two words in a sentence and provides a more flexible way to represent the context dependency features to complement BiLSTMs. Last, we concatenate the representations learned by all preceding layers as the final features for extracting aspect terms.

Methodology. Given a review sentence comprising of a sequence of tokens by $s = \{w_1, w_2, \ldots, w_T\}$, where T is the number of tokens, we aim to predict an aspect label sequence $y = \{y_1, y_2, \ldots, y_T\}$ for s. Each token w_t is classified as y_t that comes from a finite label set $Y = \{B, I, O\}$. B, I, and O represent the beginning of an aspect term, the inside of an aspect term, and non-aspect words, respectively. The specific steps of the model are described as follows.

Step 1. Double embedding mechanism. Each token w_t in the review sentence gets its two corresponding continuous representations based on two pre-trained embedding matrices. One is a general embedding x_t^g, and the other is a domain-specific embedding x_t^d. The scope of the domain embeddings coincides with the domain to which the datasets belong. Because the domain and global embeddings are trained with different datasets individually, their embedding spaces differ. To preserve their context features for label prediction, we build the initial token-level contextualized representations with the global and domain embeddings based on BiLSTMs [6], respectively. Let $\text{LSTM}^{\mathcal{S}}$ denote an LSTM unit, $\mathcal{S} \in \{G, D\}$ is the task indicator, and G and D are the notations for global and domain tasks, respectively. Below is the calculation process.

$$\overrightarrow{h_t^{\mathcal{S}}} = \text{LSTM}^{\mathcal{S}}(\overrightarrow{h_{t-1}^{\mathcal{S}}}, x_t^{\mathcal{S}}), \tag{1}$$

$$\overleftarrow{h_t^{\mathcal{S}}} = \text{LSTM}^{\mathcal{S}}(\overleftarrow{h_{t+1}^{\mathcal{S}}}, x_t^{\mathcal{S}}), \tag{2}$$

$$h_t^{\mathcal{S}} = [\overleftarrow{h_t^{\mathcal{S}}} : \overrightarrow{h_t^{\mathcal{S}}}], \tag{3}$$

where for $1 \leqslant t \leqslant T$, $x_t^{\mathcal{S}}$ is the word embedding of w_t in $\mathcal{S}$ task, and $h_t^{\mathcal{S}}$ is the hidden state at time-step t. Then, h_t^G and h_t^D are concatenated by

$$h_t^1 = [h_t^G : h_t^D], \tag{4}$$

where h_t^1 denotes the representation of the tth word in the first BiLSTM layer, and ":" indicates the concatenation operation.

* Corresponding author (email: chenchen@pku.edu.cn, wanghf@pku.edu.cn)

Step 2. Multi-layer BiLSTM network. Because the feature representations vary with network depth in a deep neural network [4], we employ multi-layer BiLSTM to learn multi-level features. The hidden state of an LSTM unit in layer $l-1$ is used as input for the LSTM unit in layer l in the same position. The input of the second layer is the output of the double embeddings module. The hidden state of the l-layer BiLSTM at t position h_t^l is calculated as

$$\overrightarrow{h_t^l} = \mathrm{LSTM}(\overrightarrow{h_{t-1}^l}, h_t^{l-1}), \tag{5}$$

$$\overleftarrow{h_t^l} = \mathrm{LSTM}(\overleftarrow{h_{t+1}^l}, h_t^{l-1}), \tag{6}$$

$$h_t^l = [\overleftarrow{h_t^l} : \overrightarrow{h_t^l}], \tag{7}$$

where $2 \leqslant l \leqslant L$ and L is the total BiLSTM layers. The model uses multiple BiLSTMs with different parameters.

Step 3. Self-attention mechanism. While BiLSTM can capture context dependencies with sequential architecture, it is not good at remembering long-term dependencies because of vanishing gradient problems. We use a self-attention mechanism to capture the context dependency information between two tokens within a sentence, regardless of their distance. As per the need of the densely-connected network, we first obtain the matrix of input vectors $V \in \mathbb{R}^{T\times d}$ by concatenating the outputs of all preceding layers in order, including the global embeddings, the domain embeddings, and the hidden states of all BiLSTM networks, where d is the dimension of the concatenated vectors. Following [7], we split the channels equally as per the number of heads denoted by m. These parallel heads are employed to focus on a different part of the channels of the value vectors. We denote the input vectors of the ith head vectors by $V_i \in \mathbb{R}^{T\times d/m}$ and control the attention calculation by m. The scaled dot-product attention is then used on the following equation:

$$p_i = \mathrm{softmax}\left[\frac{V_i V_i^{\top}}{\sqrt{d/m}}\right], \tag{8}$$

$$O_i = p_i^{\top}(V_i), \tag{9}$$

where p_i is a normalized weight matrix. Finally, the outputs $O_i \in \mathbb{R}^{T\times d/m}(1 \leqslant i \leqslant m)$ produced by parallel heads are concatenated together. Again, a linear layer is used to mix different channels from different heads, written as

$$O = \mathrm{Concat}(O_1, \ldots, O_m), \tag{10}$$

$$A = OW_{sa}, \tag{11}$$

where $W_{sa} \in \mathbb{R}^{d\times d_{\mathrm{attn}}}$ is a parameter matrix for mapping O to the output matrix $A = \{a_1, \ldots, a_T\}$, which forms a sequence of self-attentive token encoding. d_{attn} is the dimension of the self-attention output vector for each token.

Step 4. Dense connectivity. Our model employs a densely connected mechanism to generate the prediction vector f_t, which represents the feature by concatenating all other layers representations at position t. The formula is given by

$$f_t = [x_t^G : x_t^D : h_t^1 : \cdots : h_n^t : a_t], \tag{12}$$

where x_t^G and x_t^D are the global and domain embeddings of w_t, respectively, h_t^* denotes the hidden states of all BiLSTM layers at position t, and a_t is the tth output vector of the self-attention layer. The model outputs a final prediction for the token w_t writen as

$$P(y_t|w_t) = \mathrm{Softmax}(W f_t + b), \tag{13}$$

where W and b are weight and bias of the outermost full connected layer, respectively.

Experimental setup. We evaluate the model using two benchmark datasets from the SemEval ABSA challenge: the restaurant domain datasets from [8] (denoted by 16-R) and the laptop domain datasets from [9] (denoted by 14-L). We use a pre-trained embedding[1)] for the general-purpose embeddings. For domain-specific embeddings, we use embeddings trained by [3]. The out-of-vocabulary word embeddings are calculated by fastText[2)]. The number of BiLSTM layers L is 3. The dimensions of h_t^G, h_t^D, h_t^2 and h_t^3 are set to 100, 100, 200, and 200, respectively. d_{attn} is set to 400. The model is trained with stochastic gradient descent (SGD) with a learning rate of 0.07. We also employ dropout on the word embeddings and the ultimate features with a dropout rate of 0.5. All other parameters are trainable.

Results and discussion. Table 1 shows the comparison results. The results of `CMLA`, `THA_STN` and `DE_CNN` are copied from [1–3] as our comparative methods. We implement the models in the second group. `BiLSTMs_G` and `BiLSTMs_D` directly predict labels based on the output of multi-layer Bi-LSTMs. `OURS_w/o_DE` connects all layers except the embedding layer to the prediction layer. `OURS_w/o_SA` removes the self-attention component from the proposed model. `OURS_w/o_DE/SA` only connects the outputs of each BiLSTM layer for prediction.

Table 1 Experimental results (F1-score (%))

Model	14-L	16-R
`CMLA` [1]	77.80	72.77
`THA_STN` [2]	79.52	73.61
`DE_CNN` [3]	81.59	74.37
`BiLSTMs_G`	81.1	75.87
`BiLSTMs_D`	78.74	77.31
`OURS_w/o_DE`	79.78	76.01
`OURS_w/o_SA`	81.34	**78.18**
`OURS_w/o_DE/SA`	80.44	77.4
`OURS`	**81.60**	77.81

The proposed model performs the best. We highlight our main findings. (1) Compared with `BiLSTMs_G` and `BiLSTMs_D`, `OURS` improves performance because they only leverage information from the last layer of deep neural networks and could result in the loss of some features from the lower layers. (2) Compared with the state-of-the-art models, this model exhibits various performances on the two datasets. From the results of `BiLSTMs_G` and `BiLSTMs_D`, we can see that the restaurant domain embeddings achieve much better performances than the laptop domain embeddings. Thus, we suspect the improvement by incorporating the laptop-domain embeddings is limited. (3) Word embeddings contain morphological information, so adding the embedded vectors to the prediction feature helps to extract aspect terms. `OURS` performs better than `OURS_w/o_DE`, and `OURS_w/o_SA` performs better than `OURS_w/o_DE/SA`. (4) The effectiveness of

1) https://nlp.stanford.edu/projects/Glove/.
2) https://github.com/facebookresearch/fastText.

the self-attention component is closely related to word embeddings. The self-attention component seems useless for 16-R. We suspect that the self-attention mechanism is more useful for the GloVe embedding-based models. Because the domain embeddings could play an important role in the prediction on the 16-R, the self-attention features are less beneficial.

Conclusion. We present a densely-connected neural network for the aspect term extraction task. It enables preserving feature information from the bottommost layer to the uppermost layer in deep neural networks. The experiment results on two standard benchmark ABSA datasets indicate that our model improves ATE performances and leads to new advanced results.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61433015).

References

1 Wang W, Pan S J, Dahlmeier D, et al. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017. 3316–3322

2 Li X, Bing L, Li P, et al. Aspect term extraction with history attention and selective transformation. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, 2018. 4194–4200

3 Xu H, Liu B, Shu L, et al. Double embeddings and CNN-based sequence labeling for aspect extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 2: 592–598

4 Peters M E, Neumann M, Zettlemoyer L, et al. Dissecting contextual word embeddings: architecture and representation. In: Proceedings of the 2018 Conference on EMNLP, Brussels, 2018. 1499–1509

5 Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, 2017. 4700–4708

6 Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw, 2005, 18: 602–610

7 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems, Long Beach, 2017. 5998–6008

8 Pontiki M, Galanis D, Papageorgiou H, et al. Semeval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, 2016. 19–30

9 Pontiki M, Galanis D, Papageorgiou H, et al. Semeval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation, Dublin, 2015. 27–35