SC-NER: A Sequence-to-Sequence Model with Sentence Classification for Named Entity Recognition

Yu Wang, Yun Li(✉), Ziye Zhu, Bin Xia, and Zheng Liu

Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China
{2017070114,liyun,1015041217,bxia,zliu}@njupt.edu.cn

Abstract. Named Entity Recognition (NER) is a basic task in Natural Language Processing (NLP). Recently, the sequence-to-sequence (seq2seq) model has been widely used in NLP task. Different from the general NLP task, 60% sentences in the NER task do not contain entities. Traditional seq2seq method cannot address this issue effectively. To solve the aforementioned problem, we propose a novel seq2seq model, named SC-NER, for NER task. We construct a classifier between the encoder and decoder. In particular, the classifier's input is the last hidden state of the encoder. Moreover, we present the restricted beam search to improve the performance of the proposed SC-NER. To evaluate our proposed model, we construct the patent documents corpus in the communications field, and conduct experiments on it. Experimental results show that our SC-NER model achieves better performance than other baseline methods.

Keywords: Named Entity Recognition · Sequence-to-sequence model · Deep learning

1 Introduction

Named Entity Recognition (NER) [27] is a basic task in Natural Language Processing (NLP) [9], which has attracted extensive attention for a long time [3]. NER aims to identify entities (e.g., names, places, and organization names) in the text. An entity can express the core information of the sentence, which is useful for various NLP tasks [16,17].

In the last few years, deep learning has been widely used in NLP tasks such as text classification [18], language recognition [24], and machine translation [12]. In these studies, the Recurrent Neural Networks (RNNs) [22] is widely used to obtain the sequential nature of the sentence. Especially, Long Short-Term Memory network (LSTM) [24], as a popular RNN model, is adopted to extract semantic features reflecting sequential nature of the text. Moreover, the RNNs

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11439, pp. 198–209, 2019.
https://doi.org/10.1007/978-3-030-16148-4_16

like to model the conditional probability $P(y|x)$, where the output sequence $y = (y_1, ..., y_n)$ (n is the length of sequence.) is generated from the input sequence x. Recently, researchers leverage sequence-to-sequence (seq2seq) model [32] to solve the sequence generation [5] problems in NLP tasks. In general, the NER is considered as a sequence problem [11], where the NER model tags each word to indicate whether the word is part of any named entity. So we apply seq2seq model [13] to the NER task, which encodes a sequence as a vector and decodes the vector into a tag sequence. Both encoder and decoder are constructed based on RNN.

However, the most evident difference between the normal natural language generation task [30] (e.g., human-computer conversation) and the NER task [29] is that 60% sentences in our corpus may not have an entity. This feature is non-trivial, and we should pay attention to it when designing the seq2seq model in NER task. Moreover, the seq2seq model learns to predict an output sequence at the training time, and it chooses the best tag sequence using the beam search [34] at the test time. The beam search is a heuristic search algorithm [21], and it works well in many tasks with large search space. However, the standard beam search [14] is not suitable for NER task, since the search space of NER task is small.

In the proposed seq2seq model named SC-NER, the tag sequence is generated in two steps: (1) We use the encoder's output to construct a classifier, which identifies whether the input sequence has an entity. In particular, we choose the LSTM for both encoder and decoder. The encoder will output the cell state and the last hidden state. The last hidden state is input into the classifier. (2) The classifier's output is considered as the starting vector of the decoder. Usually, the decoder generates the tag sequence based on the starting vector and cell state. In addition, the beam search is always utilized in tag sequence generation. The standard beam search, however, is not suitable for NER task. Therefore, we present a restricted beam search. The main contributions of our work are as follows:

- To the best of our knowledge, it is the first time for the seq2seq model to be used in NER task.
- In the proposed SC-NER model, a classifier is added to determine whether sentences have entities. Moreover, the training of classifier, encoder, and decoder is seamless.
- We present a restricted beam search to adjust the search space. The restricted beam search is more suitable to the NER task than the standard beam search.

The remaining part of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the details of our model. Section 4 introduces the data set. Experiment results are reported in Sect. 5. Section 6 concludes our work.

2 Related Work

Su and Su [31] presented a Hidden Markov Model (HMM) based block marker to identify and classify names, times, and numbers. It achieves high performance on the MUC-6 and MUC-7 English NER tasks and performs better than manual rule-based methods. Hai and Ng [15] proposed a maximum entropy approach for NER task, which showed the feasibility of extracting useful features (global features) using the occurrences of each word in a document. Mccallum and Li [26] presented a feature induction method for CRFs (Conditional Random Fields), which obtained a high F1-score.

In order to improve the performance on NER with more diverse entity types, researchers tend to use the deep learning methods. Wu and Xu [35] used DNNs for the NER task, the experiments show that it outperforms the CRF's model. Chiu and Nichols [7] showed that CNN is effective in the feature engineering. However, the CNN pays more attention to learn the spatial features, while the text is mainly represented by temporal features. Instead of the CNN, researchers use the RNN [2] to learn the sequential feature in the text. However, Bengio et al. [4] presents that RNN did not consider the long-term dependencies, and Long Short-Term Memory network (LSTM), which is a particular RNN, is adopted to learn the long-term dependencies. Yao and Huang [36] used the LSTM for the word segmentation, and it has a positive impact on the NER task. Lin et al. [23] proposed a Multi-channel BiLSTM-CRF Model for NER in Social Media.

Recently, researchers tend to use the sequence-to-sequence model for the NLP task. Cho et al. 168 [8] proposed a novel neural network model called RNN Encoder-Decoder for statistical machine translation. The model can learn a semantically and syntactically meaningful representation of linguistic phrases. Shao et al. [30] used the sequence-to-sequence model for generating high-quality and informative conversation responses. Moreover, self-attention was added to the decoder to maintain coherence in longer responses. Konstas et al. [19] applied the sequence-to-sequence models for parsing and generation. It showed that sequence-based models are robust against ordering variations of graph-to-sequence conversions.

3 Model

In this section, we present our SC-NER model in detail. Section 3.1 provides an overview, Sect. 3.2 introduces the encoder and classifier, Sect. 3.3 elaborates the decoder. We describe the restricted beam search algorithm in Sect. 3.4.

3.1 Overview

Figure 1 illustrates the overall architecture of our model. The sentence classifier is added to the traditional seq2seq model. The traditional seq2seq model [10,32] contains the encoder and the decoder. We modify the encoder and use the last hidden state h as the input of sentence classifier. Moreover, the decoder generates the final sequence based on the cell state c and the output of the classifier (i.e., the class label l).

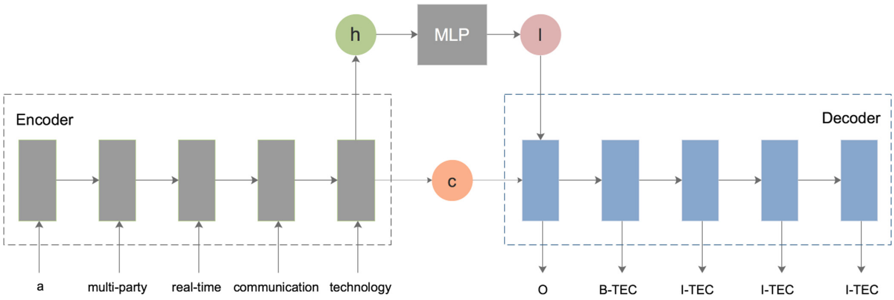

Fig. 1. An overview of our SC-NER model.

3.2 Encoder and Classifier

In the SC-NER model, we use two different LSTMs [24]: one for encoder and another for decoder. It is common to train the LSTMs on the pairs of two sequence in different features at the same time. In the NER task, the input sequence represents a natural sentence. However, the output sequence is just a set of tags without any semantic information. Moreover, attention-based LSTM [6, 33] can significantly outperform standard LSTM, then we use the LSTM with attention for encoder part.

Figure 2 shows the structure of the cell state in LSTM [2]. It uses the input gate(i_t), forget gate(f_t) and output gate(o_t) to protect and control the cell state(c_t).

$$f_t = \sigma(W_f \cdot [c_{t-1}, h_{t-1}, x_t] + b_f), \tag{1}$$

$$i_t = \sigma(W_i \cdot [c_{t-1}, h_{t-1}, x_t] + b_i), \tag{2}$$

$$\widetilde{c_t} = \tanh(W_c \cdot [c_{t-1}, h_{t-1}, x_t] + b_c), \tag{3}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \widetilde{c_t}, \tag{4}$$

$$o_t = \sigma(W_o \cdot [c_{t-1}, h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t \cdot \tanh(c_t), \tag{6}$$

where $\sigma(\cdot)$ is the logistic sigmoid function, W and b denote the weight matrix and the bias vector of each gate respectively, x_t presents the current input vector, h_{t-1} means the previous hidden state, and c_{t-1} denotes the previous cell state.

According to statistics, more than half of sentences in corpus have no entity. For example, the sentence "This method performs very well in many applications" contains no entity, and its tag sequence is very simple. Thus, the NER model should filter out sentences that do not contain entities. Therefore, we propose an Multi-Layer Perceptron (MLP) classifier to determine whether sentences contain entities.

Note that, the outputs of encoder are the last hidden state (h) and cell state (c) [30]. We can observe the difference between the hidden state and cell state from the above Eqs. 4 and 6. The LSTM uses the cell state to store the context

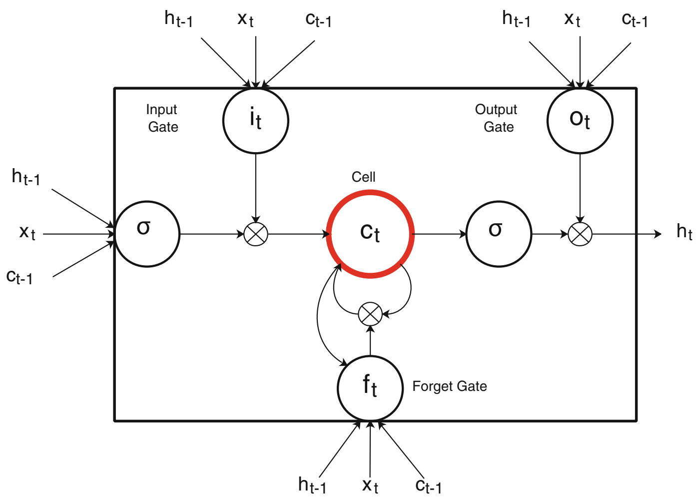

Fig. 2. The structure of the cell state in LSTM is combined with the three gates.

information. Therefore, the cell state changes slowly. The hidden state, however, changes faster than the cell state. The h_t may be very different from the h_{t-1} since Eq. 6 shows that it contains the activation function (i.e., tanh) and the dot product. Generally, the last hidden state pays more attention to the conclusion of the sentence and the cell state carries the information of the whole sentence. Therefore, the last hidden state is used as input of MLP classifier. The classifier is added after the encoder. The output of classifier based on Eq. 7 is class label l. The training objective of the classifier is to minimize the Cross-Entropy of the data.

$$l = \sigma(W_l \cdot h_t + b_l). \tag{7}$$

It is well-known that NER has the data imbalance problem. The seq2seq model with MLP classifier also can solve this problem to a certain extent.

3.3 Decoder

The decoder is also realized by LSTM. The cell state and the classifier's output are taken as the input in our SC-NER model, which is different from the traditional seq2seq model. The inputs of decoder in traditional seq2seq model are the cell state and a starting vector, and the starting vector is always initialized as the all-zero vector or the last hidden state of encoder. In other words, the output of MLP classifier is used as the starting vector in our model, which will inform the decoder whether a sentence contains entity. If there is no entity in this sentence, the decoder can almost ignore the information of the cell state.

The training method for decoder is based on SGD (Stochastic Gradient Descent) to minimize the negative logarithm likelihood.

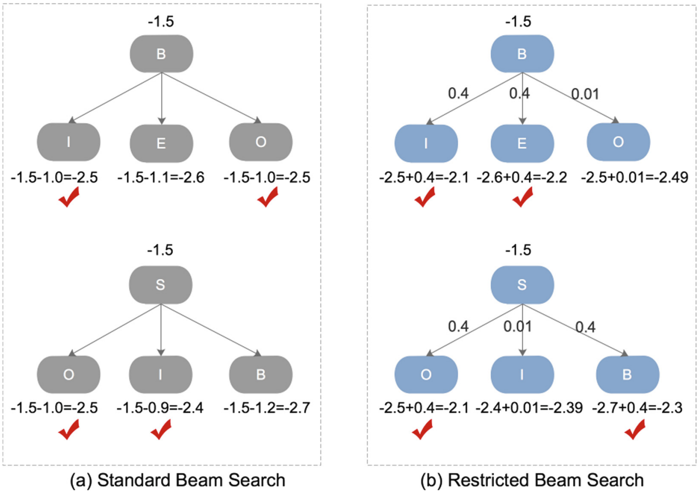

Fig. 3. The structure of two beam search algorithm.

3.4 Restricted Beam Search

The seq2seq models are trained to generate an output sequence. At the test time, the model usually uses beam search [34] to choose the best sequence given the input. Beam Search is a heuristic graph search algorithm. It is usually used when the search space of the graph is large. As illustrated in Fig. 3(a), at each step of the depth expansion, the score of each hypothesis is calculated by standard beam search. Some low score hypotheses are cut off, and the high score hypotheses are left.

Table 1. Examples of the named entities and its tags.

Named Entity	Tag
MCU	S
Vector coding	B E
Hidden Markov models	B I E

The search space, however, is relatively small in the NER task [23], where the search space (i.e. tag set) includes five tags to represent the entity boundary information i.e. BIESO [1] (e.g., Beginning, Internal, Ending, Single and Other). ‘B’, ‘I’, and ‘E’ denote the beginning, the internal and the ending of an entity. ‘S’ denotes the entities with a single word. ‘O’ denotes the other words. Table 1 shows some examples of the named entities and its tags. Therefore, the beam size for NER task is small. Moreover, we add some constraints in the restricted beam search to adjust the search process. As shown in Fig. 3(b), the score of each hypothesis is updated by adding an additional term γ to the original value, where γ denotes the dependency between the current hypothesis and its parent.

We set the term γ as the transition probability between the tags in the corpus. The γ is able to reduce the error of sequence generation. For instance, the tag sequence "O O E" is never generated by the decoder since'E' should be presented following the 'B' or 'I'. The restricted beam search thus generally favors choosing hypotheses with the higher transition probability, which leads to a more appropriate N-best list.

4 Data

We like to evaluate the SC-NER model on the patent documents corpus in the communications field. The data set contains approximately ten thousand patents and 1000 test patents downloaded from Google Patent Search[1]. The NER task is to recognize all the named entities and determine its type for each patent document. For example, given a sentence "The present invention relates to a multi-party real-time communication technology.", the NER task is to recognize the named entity "multi-party real-time communication" and determine its type. Table 2 shows the description of our corpus.

Table 2. The description of our corpus

Entity type	Total
Method	3041
Material	4827
Product	4205

4.1 Data Preprocessing

The patent documents corpus is provided in Extensible Markup Language (XML). Patent documents contain many fields. Some fields are noise in our task. Examples of such fields are country, bibliographic data, legal-status, or non-English abstracts. Therefore, in order to focus on the NER task, the following fields are chosen: Title, Abstract, Description, and Claims.

4.2 Named Entity Recognition

In the traditional area, NER is to determine whether a sentence contains a named entity and to identify its type. The entity types [27] generally include person's name, place name, and time information. However, in our task, the entity types are predefined as:

[1] https://patents.google.com.

Table 3. The results of different methods

Method	Type	Precision	Recall	F1-score
Passos et al. [28]	Method	87.48	81.35	84.30
	Material	88.23	83.33	85.71
	Product	88.02	81.55	84.66
Passos et al. [28] + artificial features	Method	87.36	83.42	85.34
	Material	88.63	85.85	87.22
	Product	88.25	84.87	86.53
Ma and Hovy [25]	Method	89.44	85.16	87.25
	Material	89.87	86.74	88.28
	Product	88.89	85.66	87.24
Lample et al. [20]	Method	90.30	**87.54**	88.90
	Material	91.01	**88.41**	89.69
	Product	91.36	87.33	**89.30**
SC-NER	Method	**92.30**	**87.54**	**89.86**
	Material	**92.71**	87.75	**90.16**
	Product	**91.86**	**86.30**	88.99

- Method Names:
  The meaning of a method is a theoretical solution to a certain problem.
- Material Names:
  Materials are those substances that humans use to make objects, devices, components and machines.
- Product Names:
  Products are tangible items that can be supplied to the market, used and consumed by people, and can meet people's needs, such as mobile phones, routers, processors, etc.

5 Experiments and Analysis

5.1 Experiments

In our experiment, we validate the performance of SC-NER model under different conditions. (1) The effect of the sentence classification module. (2) Whether the restricted beam search can improve the performance of the proposed model. Furthermore, we also compared the proposed model with many existing methods, such as Passos et al. [28], which is based on CRFs, and the deep learning methods based on the CNN [25] and LSTMs [20]. We evaluate all models performance using several criteria (e.g., Precision Rate, Recall Rate, and F1-score).

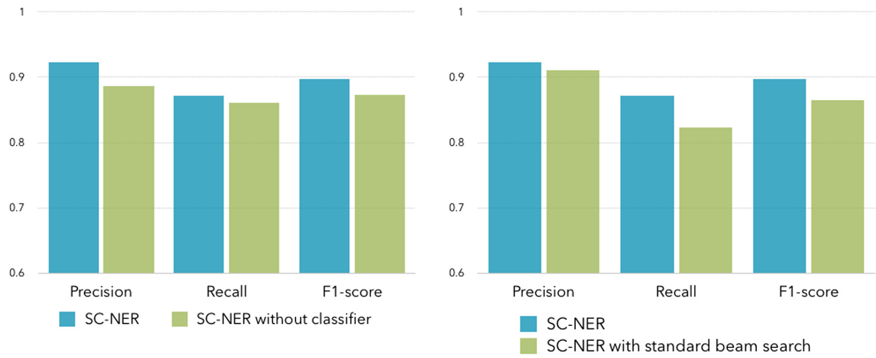

Fig. 4. The experimental results about the sentence classification module and the restricted beam search.

5.2 Results

Figure 4 shows that the performance of the SC-NER model and the SC-NER without classifier. From Fig. 4, we can observe that SC-NER achieves the better precision rate and the similar recall rate to the SC-NER without classifier. Based on the experimental result, we conclude that the sentence classifier is effective in the SC-NER model, because classifier can filter out sentences that do not contain entities.

Figure 4 also shows that the restricted beam search can improve the recall rate, and get the similar precision rate to the standard beam search. There exist two important factors that affect the recall rate. (1) Sentence contains entities, while the model can not recognize them. (2) The model tags the incorrect sequence for the sentence. The restricted beam search can alleviate the second issue above, since it can keep SC-NER model from generating the incorrect tag sequence.

Finally, Table 3 presents the results of all methods for the patent NER under precision, recall, and F1-score. The experimental result shows that the performance of our proposed model outperforms other approaches. The F1-score of the two Standford NER models are lower than SC-NER since the CRFs can not extract the effective features for patents.

As shown in Table 3, the recognition performance on different entity types are different. The performance of recognizing the Material entity is better than identifying the other types of entity, i.e., Method and Product. The reason for this result is that Material entities are usually relatively short. Method entity and Product entity appears to be compound words, which is difficult for the model to extract contextual features.

6 Conclusion

In this paper, we propose the novel seq2seq model SC-NER for the communication patent NER, and the model achieves the good performance in this NER task. We add an MLP classifier between the encoder and the decoder. The classifier can filter out sentences that do not contain entities, which can make the SC-NER model efficient and solve the data imbalance problem. Moreover, the classifier can be trained jointly with the encoder and the decoder. At the test time, we propose the restricted beam search, which is suitable for small search space in the NER task.

Experimental results show that our model makes some improvements in both the precision rate and recall rate compared with other traditional baselines such as the CRFs, CNN, and Bi-LSTMs.

In the future, we plan to enhance the generalization ability of the proposed model. The transfer learning is an option.

Acknowledgment. This work was partially supported by Natural Science Foundation of China (No. 61603197, 61772284, 61876091, 61802205), Jiangsu Provincial Natural Science Foundation of China under Grant BK20171447, Jiangsu Provincial University Natural Science Research of China under Grant 17KJB520024, the Natural Science Research Project of Jiangsu Province under Grant 18KJB520037, and Nanjing University of Posts and Telecommunications under Grant NY215045.

References

1. Deft ERE annotation guidelines: Entities. Linguistics Data Consortium (2014)
2. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 38, pp. 6645–6649. IEEE (2013)
3. Alotaibi, F., Lee, M.: A hybrid approach to features representation for fine-grained Arabic named entity recognition. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 984–995 (2014)
4. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**, 157–166 (1994)
5. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349 (2015)
6. Cheng, Y., et al.: Agreement-based joint training for bidirectional attention-based neural machine translation. arXiv preprint arXiv:1512.04650 (2015)
7. Chiu, J., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Comput. Sci. (2015)
8. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. Comput. Sci. (2014)
9. Collobert, R., Weston, J., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
10. Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016)
11. Dong, C., Zhang, J., Zong, C., Hattori, M., Di, H.: Character-based LSTM-CRF with radical-level features for Chinese named entity recognition. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC -2016. LNCS (LNAI), vol. 10102, pp. 239–250. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_20

12. Gehring, J., Auli, M., Grangier, D., Dauphin, Y.N.: A convolutional encoder model for neural machine translation. arXiv preprint arXiv:1611.02344 (2016)
13. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122 (2017)
14. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
15. Hai, L., Ng, H.: Named entity recognition with a maximum entropy approach. In: Conference on Natural Language Learning at HLT-NAACL, pp. 160–163 (2003)
16. Ji, Y., Tan, C., Martschat, S., Choi, Y., Smith, N.A.: Dynamic entity representations in neural language models. arXiv preprint arXiv:1708.00781 (2017)
17. Keith, K.A., Handler, A., Pinkham, M., Magliozzi, C., McDuffie, J., O'Connor, B.: Identifying civilians killed by police with distantly supervised entity-event extraction. arXiv preprint arXiv:1707.07086 (2017)
18. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
19. Konstas, I., Iyer, S., Yatskar, M., Choi, Y., Zettlemoyer, L.: Neural AMR: Sequence-to-sequence models for parsing and generation (2017)
20. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
21. Li, J., Monroe, W., Jurafsky, D.: A simple, fast diverse decoding algorithm for neural generation. arXiv preprint arXiv:1611.08562 (2016)
22. Li, P.H., Dong, R.P., Wang, Y.S., Chou, J.C., Ma, W.Y.: Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2664–2669 (2017)
23. Lin, B.Y., Xu, F., Luo, Z., Zhu, K.: Multi-channel BiLSTM-CRF model for emerging named entity recognition in social media. In: Proceedings of the 3rd Workshop on Noisy User-generated Text, pp. 160–165 (2017)
24. Sundermeyer, M., Ney, H., Schluter, R.: From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(3), 517–529 (2015)
25. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354 (2016)
26. Mccallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Conference on Natural Language Learning at HLT-NAACL, pp. 188–191 (2003)
27. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investig. **30**(1), 3–26 (2007)
28. Passos, A., Kumar, V., McCallum, A.: Lexicon infused phrase embeddings for named entity resolution. arXiv preprint arXiv:1404.5367 (2014)
29. Peng, N., Dredze, M.: Improving named entity recognition for Chinese social media with word segmentation representation learning. In: Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 149–155, August 2016
30. Shao, L., Gouws, S., Britz, D., Goldie, A., Strope, B.: Generating high-quality and informative conversation responses with sequence-to-sequence models (2017)
31. Su, J., Su, J.: Named entity recognition using an HMM-based chunk tagger. In: Meeting on Association for Computational Linguistics, pp. 473–480 (2002)
32. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)

33. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
34. Wiseman, S., Rush, A.M.: Sequence-to-sequence learning as beam-search optimization. arXiv preprint arXiv:1606.02960 (2016)
35. Wu, Y., Jiang, M., Lei, J., Xu, H.: Named entity recognition in Chinese clinical text using deep neural network. Stud. Health Technol. Inform. **216**, 624–628 (2015)
36. Yao, Y., Huang, Z.: Bi-directional LSTM recurrent neural network for Chinese word segmentation. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) ICONIP 2016. LNCS, vol. 9950, pp. 345–353. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46681-1_42