LOW-RESOURCE NAMED ENTITY RECOGNITION WITHOUT HUMAN ANNOTATION

Zhenshan Bao, Yuezhang Wang and Wenbo Zhang

College of Computer Science, Beijing University of Technology, Beijing, China

ABSTRACT

Most existing approaches to named entity recognition (NER) rely on a large amount of high-quality annotations or a more complete specific entity lists. However, in practice, it is very expensive to obtain manually annotated data, and the list of entities that can be used is often not comprehensive. Using the entity list to automatically annotate data is a common annotation method, but the automatically annotated data is usually not perfect under low-resource conditions, including incomplete annotation data or non-annotated data. In this paper, we propose a NER system for complex data processing, which could use an entity list containing only a few entities to obtain incomplete annotation data, and train the NER model without human annotation. Our system extracts semantic features from a small number of samples by introducing a pre-trained language model. Based on the incomplete annotations model, we relabel the data using a cross-iteration approach. We use the data filtering method to filter the training data used in the iteration process, and re-annotate the incomplete data through multiple iterations to obtain high-quality data. Each iteration will do corresponding grouping and processing according to different types of annotations, which can improve the model performance faster and reduce the number of iterations. The experimental results demonstrate that our proposed system can effectively perform low-resource NER tasks without human annotation.

KEYWORDS

Named entity recognition, Low resource natural language processing, Complex annotated data, Cross-iteration.

1. INTRODUCTION

Named entity recognition is widely applied in many scenarios, usually as a basic task for information extraction, question answering and machine translation [1]. Most existing approaches to NER focused on a supervised setup, which rely on a large amount of high-quality annotations [2]. However, in practice, it is very expensive to obtain manually annotated data. In most cases, a list of entities will be constructed to annotate the data automatically, which has high requirements for the comprehensiveness of the entity list. In some special fields, it is difficult to provide a more comprehensive list of entities. In this low-resource situation, the data automatically annotated using the entity list is often not perfect, including incomplete annotation data or non-annotated data.

Figure 1 shows an example of annotating data with a list of entities. The sentence with two named entities “Jack Davis” and “New York” of type PER (person) and LOC (location), respectively. Following the standard “BIOE” annotation system, the correct annotations is shown

David C. Wyld: CSITY, NWCOM, SIGPRO, ASOFT, AIFZ, BDIoT, ITCCMA, CLSB, DTMN, MLNLP - 2021
pp. 327-335, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111427

below the sentence. In a real scenario, the provided data annotations may be missing or incorrect, especially automatic annotations. Examples 1 and 2 (E1 and E2) are incomplete data annotations, not all entities in the corpus are annotated. The corpus with no entity annotated is non-annotated data, as shown in E3. The entity of non-annotated data may not be annotated or there may be no entity in the corpus.

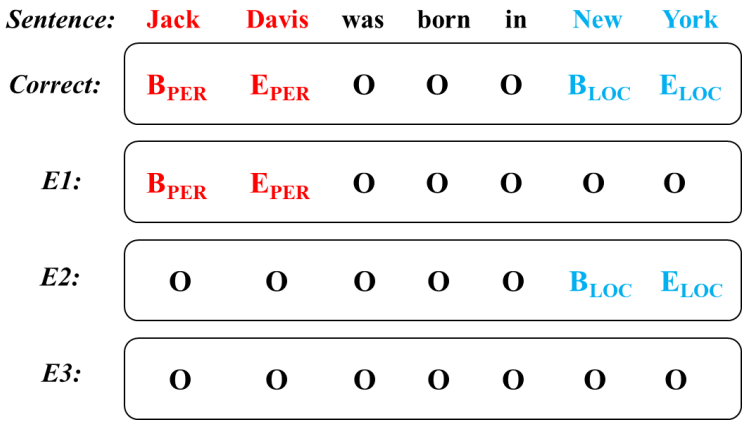

Figure 1. Example of using entity list to annotate data

It is difficult to obtain high performance when these noisy data are directly used in the training model. For these annotations, previous work mainly focused on one of these annotation data types. Using these complex data to train the NER model is a very challenging task. Solving this problem can effectively reduce the application difficulty of NER tasks in actual scenarios and reduce costs.

In this work, we present a flexible and efficient system to deal with complex data automatically annotated by entity list, and effectively use these data to improve the performance of NER model. Our system extracts semantic features from a small number of samples by introducing a pre-trained language model. Based on the incomplete annotations model [16], we relabel the data using a cross-iteration approach. We use the data filtering method to filter the training data used in the iteration process, and re-annotate the incomplete data through multiple iterations to obtain high-quality data. Finally, we use these re-annotated data to train the final NER task model. To evaluate the efficiency of our system, we conduct experiments on two real network datasets. The experimental results demonstrate that our proposed system can effectively perform low-resource NER tasks without human annotation.

2. RELATED WORK

Traditional NER methods are mainly rule-based methods and statistical-based methods. Alfred et al. and Hanisch et al. [3, 4] perform NER tasks through rule matching or grammar rules. Statistics-based methods such as Hidden Markov Model (HMM) [5], Max Entropy Model [6], Conditional Random Fields (CRF) [7] and Support Vector Machine (SVM) [8] are also classic methods for processing NER tasks. In recent years, deep learning has developed rapidly, and neural networks are usually used for NER tasks. Chiu et al. and Zhang et al. [9, 10] use neural network model to obtain character-level or word-level representations from large amounts of annotated data.

Although the named entity recognition method based on deep learning has achieved good results, most of the current deep learning models with good recognition performance often rely on a large

number of high-quality annotated data. To solve this problem, Peters et al. [11] use the pre-trained context embedding of the language model to perform the NER task through a semi-supervised method. BERT [12] shows great advantages and achieve substantial performance improvements, and pre-trained language models have become a very important component in recent. Helwe et al. [13] proposed a co-training approach, while adding Arabic-based word embedding, using a small amount of data to improve the performance of the model.

The processing of noise annotations in NER task has attracted attention. Lou et al. [14] proposed a dictionary-based graph attention model to deal with this problem. Yang et al. [15] presents an approach to utilize the data generated by distant supervision to perform newtype named entity recognition in new domains. Jie et al. [16] use re-annotation method to solve this problem, the annotations in the next iteration are re-annotated by the model learned in the previous iteration. Peng et al. [17] propose an algorithm that uses only un-annotated data and a list of named entities to process NER tasks, the PU algorithm. In order to process tokens with a variety of possible tags, AutoNER [18] proposed an improved fuzzy CRF layer for processing to improve the performance of the model.

3. APPROACH

This work is aimed to improve the performance of NER systems by inferring missing information from a small number of entity lists. We propose an efficient and flexible system to deal with complex data automatically annotated by entity list, and effectively use these data to improve the performance of NER model. Our NER system mainly includes three modules, data annotation, NER model, and cross-iteration approach using data filtering methods.

Our system does not use any human-annotated data for model training, so we use a small list of entities to automatically annotate the training data. We use the BERT-CRF model as the basic model of our NER system. Inspired by Jie et al. [16], we propose a cross-iteration approach and data filtering method to re-annotate imperfect training data. Finally, we use these re-annotated high-quality training data to train the final model to improve the performance of the NER system.

3.1. Data Annotations

For NER tasks, the most common knowledge is a list of entities describing many entities belonging to the same category. Entity lists are relatively cheap, because there are many existing lists, and if coverage requirements are not high, it is easy to create an entity list manually.

The purpose of our system is to effectively utilize the automatically annotated data under low resources, so our entity list contains only a few entities. The entity list we used only included about 30% of the entities in the unlabeled corpus. We use the obtained entity list, use the forward maximum matching algorithm, and follow the standard BIOES tagging scheme to automatically annotate the entities in the sentence. After completing the annotation, words that are not entities and entity words that are not included in the entity list are both annotated as ‘O’.

In addition, we add an extra label to each word to record whether the word is annotated as an entity by the entity list during the data annotation process. This additional label will be used in the subsequent cross-iteration approach and re-annotation process. We believe that the entities annotated with the entity list are more accurate than the iterative model predictions, so recording these words can improve the efficiency and accuracy of the cross-iteration process.

3.2. Model Architecture

The model architecture we use is a classic NER model, using the BERT [12] model to obtain word embeddings and extract features, and then use a CRF layer to obtain the output tag sequence, as shown in Figure 2.

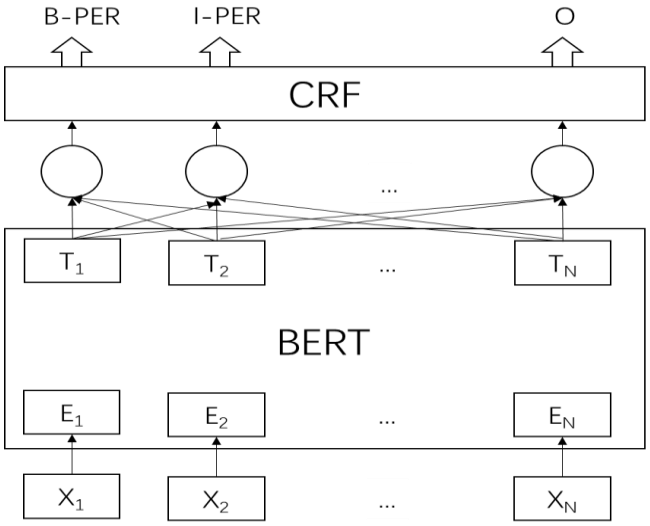

Figure 2. Model Architecture

In the task of NER, the pre-trained language model can be used to obtain the embedded representation of each word in the sentence, which can more accurately represent the semantic relationship between the entity and the context. BERT [12] is a model proposed by Google, which adopts Transformer [19] and self-attention mechanisms to learn contextual relations between words in a text, and has achieved outstanding results for a lot of NLP tasks. We use the BERT pre-training model to encode the word sequence as word embedding, and effectively mine the potential semantic information between the entity and the context, while reducing the number of samples required for training.

On top of the BERT model, a sequential CRF layer is used to perform label inference. The CRF layer can effectively constrain the dependency relationship between the predicted tags, thereby obtaining the global optimal sequence.

3.3. Cross-Iteration

Most of the annotations obtained by automatic annotation using entity list are imperfect, and direct use of these low-accuracy annotations will affect the performance of the model. Therefore, this paper proposes a cross-iteration approach and data filtering method to improve the entity annotation coverage of training data. Algorithm 1 shows the cross-iteration training procedure to re-annotate the data.

Firstly, we initialize the annotation data, record the annotation marked by entity list, and divide the training data into two folds. Then, we perform cross-iterations on the two training sets. Specifically, each time we train the model with half of the training data to predict the entity distribution of the other half of the training data. We re-annotated the imperfect annotations based on the predicted results. In order to improve the accuracy of re-annotated entities, we only re-annotate entities that were originally labeled as 'O'. At the same time, we reset the parameters of

the model before each iteration to avoid accumulating more errors in the re-annotation process. After several iterations, the coverage of entities in the training data is improved and the entity information is enriched. Finally, use these re-annotated training data to train the final model.

In the iterative process, we find that the non-annotated training data has little contribution to the prediction. Therefore, we used the data filtering method to regroup and filter the data used to train the prediction model. In the data initialization phase, we group the training data according to whether the data is non-annotated data. On this basis, each group is equally divided into two folds for cross-iteration. In addition, filter out the non-annotated data in this fold before each iteration for the training of the prediction model.

When training the final model with the re-annotated data, we do not use the data filtering method. This is because the data has been more accurately annotated during the cross-iteration process, so the non-annotated data can be used to train the model.

Algorithm 1: Cross-Iteration Training Procedure.

Input:
N: number of iterations; D_i: incompletely annotated data; D_n: non-annotated data; M: model
Output:
D_r: re-annotated training data
1: Initialize the annotation data, record the annotation marked by entity list;
2: Divide D_i into two folds D_{ia} and D_{ib};
3: Divide D_n into two folds D_{na} and D_{nb};
4: Save initial parameters of model M as M_{init};
5: for iteration=1, 2, 3 … to N do
6: Filter out the non-annotated data after relabeling in D_{na}, and merge with D_{ia} to get training data T_a
7: Filter out the non-annotated data after relabeling in D_{nb}, and merge with D_{ib} to get training data T_b
8: Reset the parameters of the model M to M_{init};
9: Train model M with T_a, get model M_a;
10: Train model M with T_b, get model M_b;
11: Use model M_a to predict the D_{ib} and D_{nb};
12: Use model M_b to predict the D_{ia} and D_{na};
13: Re-annotate D_i and D_n according to the prediction results, get re-annotated training data D_r;
14: end for

4. EXPERIMENTATION

4.1. Dataset and Experimental Settings

To evaluate the efficiency of our system, we conduct experiments on two real network datasets, AutoIE [21] and E-commerce-NER [22]. The corpus of the AutoIE dataset is derived from the title text of Youku video, which contains 10,000 samples without annotations for training and 1000 samples with fully annotated for testing. Besides the corpus, three lists of interested entity types are provided. These entities may cover around 30% entities occurring in the unlabeled corpus. The E-commerce-NER data set is a dataset crawled through the web and manually annotated. It contains two types of entities, namely products and brands. We randomly selected 30% of the annotated entities as the entity list, deleted all the original entity annotations, and re-annotated the training data with the entity list. Table 1 shows the distribution of non-annotations and incomplete annotations after using the entity list annotation.

Table 1. Statistic of dataset

Datasets	Non-annotations	Incomplete annotations	Test
AutoIE	5667	4333	1000
E-commerce-NER	574	3415	498

We compare our approach with the following model in NER task with incomplete annotation:

- Origin. It is a model that directly uses the data annotated by entity list for training;
- O-Filter. Filter out non-annotated data and use it to train the model. This model is used to evaluate the effectiveness of data filtering method.
- Baseline Jie et al. [16]: The system achieves state of art result for incomplete annotations problem in NER application, and it is employed as the baseline system for our evaluation.

We use the BERT pre-trained model "chinese_wwm_ext" which released by Cui [20]. Adam optimizer is used with the learning rate of 1e-3, and set batch size as 128. The epoch of each iteration prediction model is set to 20, and the number of iterations is set to 30.

4.2. Results

The best F1 results achieved by different methods on different datasets are listed in Table 2.

Table 2. Performance comparison between different methods

Datasets	Model	Precision	Recall	F1 score
AutoIE	Origin	0.7670	0.2981	0.4293
	O-Filter	0.7435	0.5577	0.6373
	Baseline	0.6435	0.6680	0.6555
	Ours	0.7436	0.7902	0.7662
E-commerce-NER	Origin	0.6121	0.5006	0.5508
	O-Filter	0.6107	0.5785	0.5942
	Baseline	0.5916	0.8044	0.6818
	Ours	0.6068	0.8391	0.7043

As shown in Table 2, our model can achieve the best performance on both datasets. The Origin model, which directly uses the data annotated by entity list for training, has poor performance. It is mainly due to the noise effect of a large number of non-annotated data in the training data. Therefore, we filter out non-annotated data for training. From the results of the O-Filter model, we can see that the performance of the model has been improved. Compared with the baseline model, our proposed method has achieved higher performance. On the one hand, our method uses a cross-iteration approach, which effectively utilizes the existing entity list and the semantic features of the entities in the corpus, which enhances the coverage and diversity of the entities in the training data. On the other hand, our data filtering method is used in each iteration process, reducing the accumulation of noise and further improving the accuracy of the prediction results.

On the AutoIE dataset, the best F1 score obtained by our method can reach 0.7662, which is 0.1107 higher than the baseline model. Compared with the baseline model for incomplete

annotations, our method significantly improves the precision and recall, reaching 0.7436 and 0.7902, respectively, an increase of 0.1001 and 0.1222.

On the E-commerce-NER dataset, the best F1 score can reach 0.7043, which is also higher than other methods, but the performance improvement is not obvious on the AutoIE dataset. This is because in the E-commerce-NER dataset, the proportion of non-annotated data is relatively small, and the improvement of model performance by data filtering methods is also reduced.

In order to further explore the influence of data filtering method on the iteration approach, we draw the performance curve of the test model when iterating on the AutoIE dataset, as shown in Figure 3. The test model is a model trained using all the re-annotated data after each iteration.

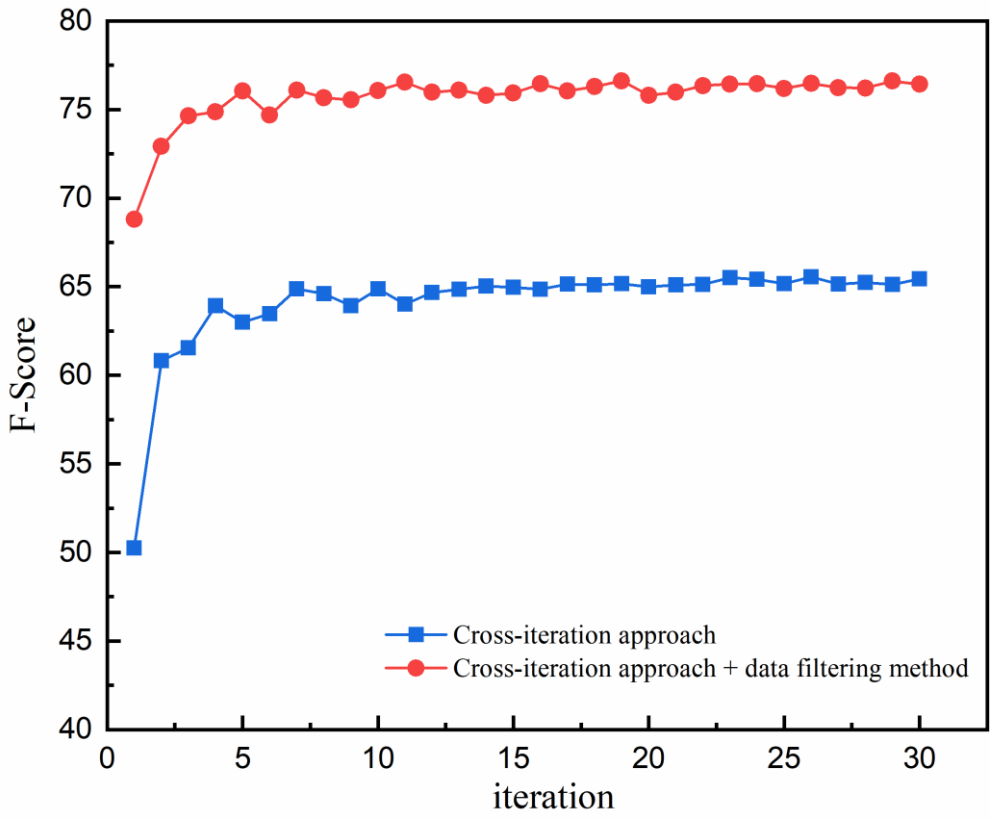

Figure 3. F1 score of iterative process on AutoIE dataset

We can see that the cross-iteration approach using the data filtering method can improve the model performance faster. For the training data automatically annotated with the entity list, the data filtering method can effectively filter out the noisy data, and at the same time re-add them to the training process through multiple cross-iterations to effectively use the data. By combining cross-iteration approach with data filtering method, we can get a higher performance model and greatly reduce the number of iterations required.

5. CONCLUSIONS

In this work, we designed an efficient and flexible system that uses automatic annotation data for NER tasks in a low-resource environment. We propose a cross-iteration approach and data filtering method to improve the entity annotation coverage of training data. Each iteration will do

corresponding grouping and processing according to different types of annotations, which can improve the model performance faster and reduce the number of iterations. Experiments on real datasets show that our system significantly improves the performance of the NER model in the case of complex annotations. In future work, we will try to filter these complex training data in more detail, and we believe that our method can also be used for more routine tasks besides sequence annotation.

ACKNOWLEDGEMENTS

This work is supported by National Key R&D Program of China (No. 2017YFC0803300), Beijing Natural Science Foundation under Grant 4172004, Beijing Municipal Education Commission Science and Technology Program under grant number KM201910005027.

REFERENCES

[1] Yadav, V., & Bethard, S. (2018, August). A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 2145-2158).

[2] Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering.

[3] Alfred, R., Leong, L. C., On, C. K., Anthony, P., Fun, T. S., Razali, M. N. B., & Hijazi, M. H. A. (2013, December). A rule-based named-entity recognition for malay articles. In International Conference on Advanced Data Mining and Applications (pp. 288-299). Springer, Berlin, Heidelberg.

[4] Hanisch, D., Fundel, K., Mevissen, H. T., Zimmer, R., & Fluck, J. (2005). ProMiner: rule-based protein and gene entity recognition. BMC bioinformatics, 6(1), 1-9.

[5] Morwal, S., Jahan, N., & Chopra, D. (2012). Named entity recognition using hidden Markov model (HMM). International Journal on Natural Language Computing (IJNLC), 1(4), 15-23.

[6] Curran, J. R., & Clark, S. (2003). Language independent NER using a maximum entropy tagger. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 (pp. 164-167).

[7] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[8] Ju, Z., Wang, J., & Zhu, F. (2011, May). Named entity recognition from biomedical text using SVM. In 2011 5th international conference on bioinformatics and biomedical engineering (pp. 1-4). IEEE.

[9] Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 4, 357-370.

[10] Zhang, Y., & Yang, J. (2018, July). Chinese NER Using Lattice LSTM. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1554-1564).

[11] Peters, M., Ammar, W., Bhagavatula, C., & Power, R. (2017, July). Semi-supervised sequence tagging with bidirectional language models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1756-1765).

[12] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).

[13] Helwe, C., & Elbassuoni, S. (2019). Arabic named entity recognition via deep co-learning. Artificial Intelligence Review, 52(1), 197-215.

[14] Lou, Y., Qian, T., Li, F., & Ji, D. (2020). A Graph Attention Model for Dictionary-Guided Named Entity Recognition. IEEE Access, 8, 71584-71592.

[15] Yang, Y., Chen, W., Li, Z., He, Z., & Zhang, M. (2018, August). Distantly supervised ner with partial annotation learning and reinforcement learning. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 2159-2169).

[16] Jie, Z., Xie, P., Lu, W., Ding, R., & Li, L. (2019, June). Better modeling of incomplete annotations for named entity recognition. In Proceedings of the 2019 Conference of the North American Chapter

of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 729-734).
[17] Peng, M., Xing, X., Zhang, Q., Fu, J., & Huang, X. J. (2019, July). Distantly Supervised Named Entity Recognition using Positive-Unlabeled Learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2409-2419).
[18] Shang, J., Liu, L., Gu, X., Ren, X., Ren, T., & Han, J. (2018). Learning Named Entity Tagger using Domain-Specific Dictionary. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 2054-2064).
[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[20] Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., & Hu, G. (2019). Pre-training with whole word masking for chinese bert. arXiv preprint arXiv:1906.08101.
[21] Yang, X., Wu, B., Jie, Z., & Liu, Y. (2020, October). Overview of the NLPCC 2020 Shared Task: AutoIE. In CCF International Conference on Natural Language Processing and Chinese Computing (pp. 558-566). Springer, Cham.
[22] Ding, R., Xie, P., Zhang, X., Lu, W., Li, L., & Si, L. (2019, July). A neural multi-digraph model for Chinese NER with gazetteers. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 1462-1467).

AUTHORS

Bao Zhenshan is an associate professor in faculty of information technology (FIT), Beijing University of Technology (BJUT), Beijing, China. His research interests include machine learning, natural language processing and their applications.

Wang Yuezhang received the Bachelor degree of Computer Science and Technology in BJUT in 2018. Now he is the postgraduate student majoring in computer Technology in BJUT. His current research is about deep learning, natural language processing, and low-resource named entity recognition.

Zhang Wenbo received her Ph.D. degree of Computer Science and technology in 2015. Now she is a lecturer in FIT, BJUT, Beijing, China. Her research interests include natural language processing, heterogeneous computing, intelligent computing system and their applications. And she is the corresponding author.