## Janko at SemEval-2023 Task 2: Bidirectional LSTM Model Based on Pre-training for Chinese Named Entity Recognition

Jiankuo Li Yunnan University janko0329@163.com Zhengyi Guan Yunnan University 1941028528@qq.com Haiyan Ding\* Yunnan University teidhy@163.com

## Abstract

This paper describes the method we submitted as the Janko team in the SemEval-2023 Task 2, Multilingual Complex Named Entity Recognition (MultiCoNER 2). We only participated in the Chinese track. In this paper, we implement the BERT-BiLSTM-RDrop model. We use the fine-tuned BERT models, take the output of BERT as the input of the BiLSTM network, and finally use R-Drop technology to optimize the loss function. Our submission achieved a macro-averaged F1 score of 0.579 on the test set.

## 1 Introduction

Named entity recognition (NER) is to identify entities with specific meanings from a given text. Named entities generally refer to entities with a specific meaning or strong referential meaning in the text, usually including personal names, place names, organizational structure names, etc. NER is a key task in natural language processing, and it is very useful in NLP tasks such as information extraction (Toda and Kataoka, 2005), question answering system (Mollá et al., 2006), and machine translation (Babych and Hartley, 2003). However, due to the nature of the language, NER is also a challenging problem. First, a word often has multiple meanings, leading to different categories of entities in different sentences. For example, Washington can mean either a place name or a person's name. Secondly, the boundary of the named entity is vague, and its forms are various. It can be a simple noun, a gerund phrase, or even a complete sentence.

MultiCoNER 2 (Fetahu et al., 2023b) aims to identify named entities with finer precision, and the organizers divided the whole competition into 13 tracks based on language. Each data set (Fetahu et al., 2023a) is composed of a training set, a validation set, and a test set. The training and validation set provides labels for named entities, and the test set provides text only. Compared with MultiCoNER (Malmasi et al., 2022b), the number of entity tags has been increased from 6 (Malmasi et al., 2022a) to more than 30, while the semantics are vaguer and the text length is shorter. In addition, mock errors are added to the test set to make the task more realistic and difficult. Similar to other NER tasks, MultiCoNER 2 uses B-I-O tags to label the tokens, where "B" indicates the first token of a concept, "I" indicates tokens inside of a concept, and "O" indicates tokens that don't belong to any entity.

This paper explains our submission to the Chinese track in MultiCoNER 2. We implemented a simple neural network system based on a pretrained language model. Our submission ranked 14th among 21 teams with 0.579 F1 score in the test set. This paper is organized as follows: In Section 2, we will briefly introduce the research status of the NER task in Chinese. Section 3 describes the models and methods used. Section 4 introduces the implementation details and experimental results. Finally, this paper is summarized in Section 5.

## 2 Related Work

English NER task is based on characters, but the Chinese NER task is more difficult than the English NER task because the meaning of words is much larger than a single word and Chinese has no space and the basic unit is the word(Geng et al., 2022). The development of NER in Chinese can be roughly divided into three categories: rule-based approach, traditional machine learning approach, and deep learning approach.

#### 2.1 Method based on rules

In a rule-based NER system, rules can be artificially designed based on domain-specific gazetteers and syntactic patterns (Krupke and Hausman, 1998; Humphreys et al., 1995; Mikheev et al., 1998). This method performs well when the vocabulary is sufficient. However, due to the characteristics of the Chinese language, the rule-based NER method is time-consuming and can not exhaust the rules and patterns in Chinese entities. In practical application, the recall rate is low and the effect is poor. Moreover, a specific system can only be applied to a specific domain and has no generalization ability.

# 2.2 Method based on traditional machine learning

Under this category, there are two methods: unsupervised learning and supervised learning. A typical approach to unsupervised learning is timeclustering, a cluster-based NER system (Collins and Singer, 1999; Nadeau et al., 2006) obtained different clusters according to text similarity, and extracted related entities by representing different entity category groups through clusters.

Supervised learning converts NER into a multiclass classification task or a sequence labeling task. Given annotated data samples, machine learning algorithms are used to learn the features of training samples, and to identify similar patterns from unknown data. Common NER models of supervised learning include HMM (Bikel et al., 1999), C4.5 decision Tree (Sekine et al., 1998), MEM (Borthwick, 1999) and CRF (Lafferty et al., 2002).

#### 2.3 Method based on deep learning

Deep learning has attracted a lot of attention for its success in various fields, and quite a few studies have applied deep learning to NER over the past few years (Guan and Liu, 2021; Mai and Zhou, 2022; Zhu and Wang, 2022; Ma et al., 2022). Compared with traditional linear models, deep learning-based models can learn complex and potential features from data through nonlinear activation functions. Therefore, it has become a trend to use deep learning models such as BiLSTM (Wei et al., 2016) and BERT (Devlin et al., 2018) to deal with NER problems in Chinese and remarkable results have been achieved.

For example, Huang et al. (2015) for the first time, applied the BiLSTM-CRF model to the NER task. In this model, first of all, the word is mapped to a word vector, and the scoring probability of each word corresponding to each label is obtained by learning context information from the BiLSTM layer. By learning the sequence-dependent information between tags, the final prediction results are obtained. Lattice-LSTM (Zhang and Yang, 2018) model encodes all words matched by a single character in a sentence as directed acyclic graphs. There is no word segmentation error in this model, and the Lattice-LSTM model has achieved good results on each data set. However, Lattice-LSTM may encounter conflicts when using external dictionaries. LR-CNN (Gui et al., 2019) model proposes to deal with such conflicts by using advanced semantics to narrow the weight of words. TENER (Yan et al., 2019) model improves location coding and self-attention in the transformer. By replacing absolute position coding with relative position coding and improving attention allocation, the transformerbased model can improve performance in the NER task and obtain excellent results. Li et al. (2020) proposed a FLAT model containing lexical information, which converted the Lattice structure into a set of fragments and incorporated trainable relative position coding.

Based on the above results, we first use BERT pre-training model to obtain word embeddings. To better obtain the scoring probability of the word corresponding to each label, we used bidirectional LSTM to obtain the context information. Finally, we used the R-Drop technique to compensate for the inconsistency of Dropout during training and testing.

## 3 Methodology

In this section, we will introduce the system architecture we use in the Chinese NER task. The system will be composed of three parts. The model we will eventually use to predict the test set files is called BERT-BiLSTM-RDrop.The architecture of the BERT-BiLSTM-RDrop model is shown in Figure 1.

## 3.1 BERT Embeddings

Transformer is an encoder-decoder architecture. BERT is a bidirectional encoder based on the transformer that uses an attention mechanism to extract information from the surrounding context. Models based on transformer architecture (Vaswani et al., 2017) have achieved very good performance in NER tasks. The Bert-based model uses only one of the encoders and is pre-trained using masking language modeling (MLM) (Taylor, 1953) and next sentence prediction (NSP) tasks. In this task, we treat the last layer of BERT as an embedding of text. In this task, we treat the last layer of BERT as an embedding of text.



Figure 1: The architecture of the BERT-BiLSTM-RDrop model.

#### 3.2 BiLSTM Neural Network

LSTM (Hochreiter and Schmidhuber, 1997) is a kind of RNN, which is very suitable for modeling sequential data, but it cannot encode information from back to front. BiLSTM is a combination of forward LSTM and backward LSTM, with one forward processing input sequence and the other reverse processing sequence. After the processing is completed, the output of the two LSTMs is joined together. BiLSTM does a better job of capturing bidirectional semantic dependencies when classifying at a finer level, especially when the output at the current moment is not only related to previous states but may also be related to future states.

## 3.3 R-Drop

Dropout (Srivastava et al., 2014) is a regularization technology widely used for deep learning. It randomly ignores or blocks some neurons in a certain proportion during training so that the model is different each time the prediction is made, which can solve the problem of overfitting. R-Drop (Liang et al., 2021) strengthens the robustness of the model to Dropout by adding a regularization term. R-Drop acts on the output layer of the model to make up for the inconsistency of Dropout during training and testing and can be widely applied to training different types of deep models. During training, R-Drop will make the same input Dropout twice. Due to the characteristic of Dropout, two different output distributions will be obtained. R-Drop operation attempts to minimize the Kullback-Leibler (KL) divergence between these two output distributions to regularize model prediction, which is:

$$\mathcal{L}_{KL}^{i} = \frac{1}{2} \left( \mathcal{D}_{KL} \left( \mathcal{P}_{1}^{w} \left( y_{i} \mid x_{i} \right) \| \mathcal{P}_{2}^{w} \left( y_{i} \mid x_{i} \right) \right) + \mathcal{D}_{KL} \left( \mathcal{P}_{2}^{w} \left( y_{i} \mid x_{i} \right) \right)$$
(1)

The loss function from the two Dropout is:

$$\mathcal{L}_{NLL}^{i} = -\log \mathcal{P}_{1}^{w} \left( y_{i} \mid x_{i} \right) - \log \mathcal{P}_{2}^{w} \left( y_{i} \mid x_{i} \right)$$
(2)

The final loss function is the sum of the two:

$$\mathcal{L}^{i} = \mathcal{L}^{i}_{NLL} + \alpha \cdot \mathcal{L}^{i}_{KL} \tag{3}$$

## 4 Results

We used the Pytorch and Huggingface libraries to implement all the models in our experiment. Experiments were performed on a machine equipped with Intel<sup>®</sup> i7-12700F CPU, Nvidia<sup>®</sup> 3080 GPU, and 32 GB of RAM running Windows 10. The training time of the BERT-BiLSTM-RDrop model is about 4 hours. We used BERT-base-chinese as the training model. The results are evaluated by F1 scores:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$
(6)

During the training process, the BERT-BiLSTM-RDrop model trains the data set provided by the organizer and saves the parameters of the 50th epoch for the subsequent test set prediction. The results are shown in Table 1. All results are final submissions to the system.

Model	Precision	Recall	F1
BERT	0.5329	0.5881	0.5514
BERT-BiLSTM	0.5458	0.6067	0.5683
BERT-RDrop	0.5305	0.6047	0.5578
BERT-BiLSTM-RDrop	0.5513	0.6251	0.5790

Table 1: Results of Chinese track

An ablation experiment was performed as shown in Table 1. It can be seen that BiLSTM can bring a large improvement because BiLSTM allows the network to obtain both forward and backward information about character sequences at each time step. Limited improvement from RDrop is expected since RDrop only reinforces the robustness of the model to Dropout by making up for inconsistencies in training and testing.

The initial learning rate was set to 1e-5 with a batch size of 4 and the  $\alpha$  hyperparameter was set to 1. The hyperparameters used are shown in Table 2.

Parameters	Chinese track
Epochs	50
Batch size	4
Initial lr	1e-5
Optimizer	Adam
Drop rate	0.1

Table 2: Hyperparameters

We tried some popular models in the verification set, and the results are shown in Table 3. It can be seen that the BERT-BiLSTM-RDrop has a better effect. We also tried other Bert-based models, such as BERT-base-uncased, RoBERTa-base, and XLM-RoBERTa-base, and surprisingly, the results were not as good as BERT-base-chinese. In addition, we tried to use the integrated network of BiLSTM and CNN to replace the single BiLSTM and average method of voting method selection, but it is a pity that there is no significant improvement in measurement.

Model	F1
RoBERTa-BiLSTM-CRF	0.7297
RoBERTa-BiLSTM-RDrop	0.7343
BERT-BiLSTM+CNN	0.7240
BERT-BiLSTM-CRF	0.7360
BERT-BiLSTM-RDrop	0.7452

Table 3: Results of Different Models

The best test set prediction submitted by our team was generated by the BERT-BiLSTM-RDrop model. Finally, Janko team ranked 14 out of 21 teams with a 0.579 F1 score.

#### 5 Conclusion

In this paper, we described our method, which is based on the pre-trained BERT model and is used to predict the NER task SemEval 2023 Task 2: Multilingual Complex Named Entity Recognition. We implemented the BERT-BiLSTM-RDrop model and prove the validity of the different model compositions. Our submission achieved a macro-averaged F1 score of 0.579 on the test set. In the future, we will continue to improve the system, use more advanced models, and try model integration methods.

## References

- Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003.*
- D. M. Bikel, R. Schwartz, and R. M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, 34.
- Andrew Borthwick. 1999. A Maximum Entropy Approach to Named Entity Recognition. Ph.D. thesis, New York University, USA.
- M. Collins and Y. Singer. 1999. Unsupervised Models for Named Entity Classification. In *Empirical Methods in Natural Language Processing*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.
- Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. Multi-CoNER v2: a Large Multilingual dataset for Finegrained and Noisy Named Entity Recognition.
- Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023b. SemEval-2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2). In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023). Association for Computational Linguistics.
- Zhichao Geng, Hang Yan, Zhangyue Yin, Chenxin An, and Xipeng Qiu. 2022. TURNER: The Uncertaintybased Retrieval Framework for Chinese NER. *CoRR*, abs/2202.09022.
- Zhengyi Guan and Renyuan Liu. 2021. Yunnan-Deep at ehealth-KD Challenge 2021: Deep Learning Model for Entity Recognition in Spanish Documents. In *IberLEF@ SEPLN*, pages 731–736.
- Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019. CNN-Based Chinese NER with Lexicon Rethinking. In *International Joint Conference on Artificial Intelligence*.

- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9:1735– 1780.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR*, abs/1508.01991.
- K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, and Y. Wilks. 1995. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In *Conference on Message Understanding*.
- G. Krupke and K. Hausman. 1998. Isoquest Inc: description of the NetOwl(TM) extractor system as used for MUC7.
- J. Lafferty, A. Mccallum, and Fcn Pereira. 2002. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *proceedings of icml*.
- Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. *CoRR*, abs/2004.11795.
- Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-Drop: Regularized Dropout for Neural Networks. In Advances in Neural Information Processing Systems, volume 34, pages 10890–10905. Curran Associates, Inc.
- Hangchao Ma, You Zhang, and Jin Wang. 2022. Pretrained Models with Adversarial Training for Named Entity Recognition in Scientific Text. In 2022 International Conference on Asian Language Processing (IALP), pages 259–264. IEEE.
- Hanjie Mai and Xiaobing Zhou. 2022. Clinical Text Entity Recognition Based on Pretrained Model and BiGRU-CRF.
- Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. MultiCoNER: A large-scale multilingual dataset for complex named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1412–1437, Seattle, United States. Association for Computational Linguistics.
- A. Mikheev, C. Grover, and M. Moens. 1998. Description of the LTG system used for MUC-7.
- Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.

- David Nadeau, Peter D. Turney, and Stan Matwin. 2006. Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity. In Advances in Artificial Intelligence, 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006, Proceedings, pages 266–277.
- Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A decision tree method for finding and classifying names in Japanese texts. In *Sixth Workshop on Very Large Corpora*.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929– 1958.
- Wilson L. Taylor. 1953. "Cloze Procedure": A New Tool for Measuring Readability. *Journalism & Mass Communication Quarterly*, 30:415 – 433.
- H. Toda and R. Kataoka. 2005. A search result clustering method using informatively named entities. In *Web Information and Data Management*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.
- Qikang Wei, Tao Chen, Ruifeng Xu, Yulan He, and Lin Gui. 2016. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*, 2016. Baw140.
- Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. CoRR, abs/1911.04474.
- Yue Zhang and Jie Yang. 2018. Chinese NER using Lattice LSTM. *ArXiv*, abs/1805.02023.
- Zhengzhong Zhu and Liqing Wang. 2022. BERT-BiLSTM Model for Entity Recognition in Clinical Text. In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022, volume 3202 of CEUR Workshop Proceedings. CEUR-WS.org.