NetEase.AI at SemEval-2023 Task 2: Enhancing Complex Named Entities Recognition in Noisy Scenarios via Text Error Correction and External Knowledge

Ruixuan Lu[1†], Zihang Tang[1‡], Guanglong Hu[2†], Dong Liu[2†], Jiacheng Li[*†]

[†]NetEase Digital Sail, NetEase Inc.

[‡]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China

{luruixuan,hzhuguanglong,hzliudong,lijiacheng01}@corp.netease.com

zihangt@mail.ustc.edu.cn

Abstract

Complex named entities (NE), like the titles of creative works, are not simple nouns and pose challenges for NER systems. In the SemEval 2023, Task 2: MultiCoNER II was proposed, whose goal is to recognize complex entities against out of knowledge-base entities and noisy scenarios. To address the challenges posed by MultiCoNER II, our team NetEase.AI proposed an entity recognition system that integrates text error correction system and external knowledge, which can recognize entities in scenes that contain entities out of knowledge base and text with noise. Upon receiving an input sentence, our systems will correct the sentence, extract the entities in the sentence as candidate set using the entity recognition model that incorporates the gazetteer information, and then use the external knowledge to classify the candidate entities to obtain entity type features. Finally, our system fused the multi-dimensional features of the candidate entities into a stacking model, which was used to select the correct entities from the candidate set as the final output. Our system exhibited good noise resistance and excellent entity recognition performance, resulting in our team's first place victory in the Chinese track of MultiCoNER II.

1 Introduction

Named Entity Recognition (NER) is a core Natural Language Processing (NLP) task that aims to detect and classify entities in a sequence. In practical and open-domain settings, however, it is difficult for machines to process complex and ambiguous named entities(Ashwini and Choi, 2014), and the performance of the NER system can be significantly reduced when the text contains noise (such as spelling errors). To alleviate this issue, SemEval 2023 Task 2(Fetahu et al., 2023b,a; Malmasi et al., 2022b,a) was formulated to focus on recognizing semantically ambiguous and complex entities against out of knowledge-base entities and noisy scenarios. In practice, noise in the text leads to deviation of the text semantics. Therefore, it is often necessary to employ some methods to correct the erroneous text in order to minimize the semantic distortion and its impact on the textual understanding. When labeling complex entities, people often rely on external knowledge to aid them. Several studies have found that incorporating prior knowledge is beneficial for entity recognition (Tedeschi et al., 2021; Hu and Wei, 2020; Seyler et al., 2018). Therefore, we believe that correcting noise and leveraging external knowledge can help to recognize entities in noisy text. To effectively combine all the features, we use a stacking model (Pavlyshenko, 2018) to integrate the semantic features of the text and external knowledge features for entity recognition.

In this article, our system consists of multiple modules. After the text is inputted, the text error correction module is used to eliminate noise. Then, the entity recognition module performs the entity recognition task on the corrected text and generates the span and category of the candidate entity set. Subsequently, the entity span is fed into the knowledge-based classifier module to gain the entity related information from the knowledge system and obtain the entity type by the classifier. Finally, the candidate entity span and entity type predicted by multiple modules are inputted into the stacking model to comprehensively assess the correctness of the entity and output the correct entity judged by the model.

2 Related Work

NER(Sundheim, 1995) is a basic task in natural language processing. This task has many applications in social media(Derczynski et al., 2017), news(Sang and Erik, 2002; Sang and De Meulder, 2003), E-commerce(Fetahu et al., 2021; Wang et al., 2021) and medical field(Doğan et al., 2014;

[1]:First authors,[2]:Second authors,[*]:Corresponding authors.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 897–904
July 13-14, 2023 ©2023 Association for Computational Linguistics

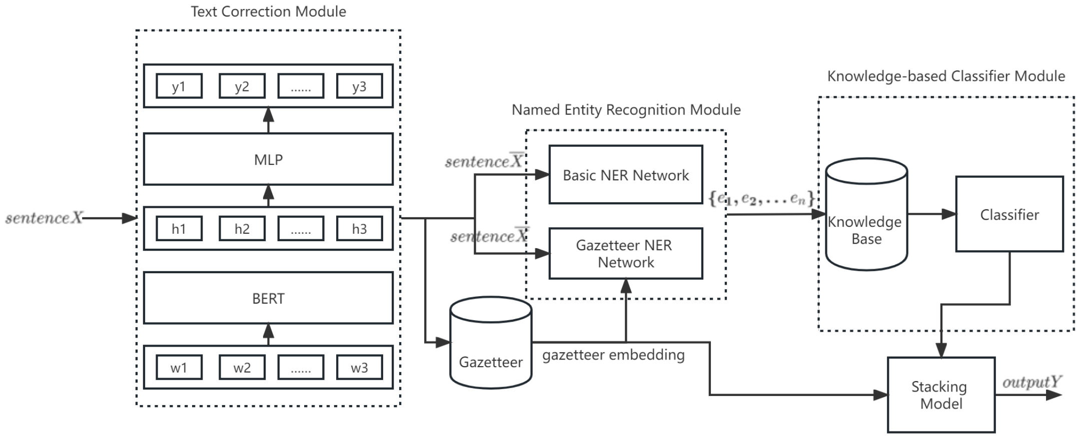

Figure 1: An image of system architecture

Li et al., 2016) and other fields. Recently, models such as BERT(Devlin et al., 2018), Tacl-BERT(Su et al., 2021), Nezha(Wei et al., 2019) and other models that contain context information based on pre-training have significantly improved the performance of NER. Recent works (Ilić et al., 2018; Akbik et al., 2018; Straková et al., 2019) have focused on combining different forms of training to enhance the performance of token embeddings. Subsequently, many studies have demonstrated that document-level context can significantly improve token representation, resulting in state-of-the-art performance (Yu et al., 2020; Luoma and Pyysalo, 2020; Yamada et al., 2020; Wang et al., 2020). In SemEval 2022 Task 11, two teams achieved impressive results. The first team proposed a knowledge-based NER system (Wang et al., 2022), while the second team utilized a gazetteer constructed from Wikidata (Chen et al., 2022) to enhance their NER model's performance. These methods have achieved top-ranking performance in multiple tracks.

3 Our System

This section will introduce the various parts of our system and how the system works. Figure 1 shows composition of our system.

3.1 Text Correction Module

The text correction module is designed to identify and correct the noise in text, such as spelling errors. The input of the module is noisy text, and the output is the modified, corrected text. To facilitate the error correction process, a label system is designed for text error correction, containing two types of labels, namely, replacement label and correct label. Replace labels of the form DELETE|X, where X is a character, are equal in number to the character set in the dictionary, meaning the token corresponding to the sentence is to be replaced with X. The correct label indicates that the token in the sentence is correct and does not need to be modified. We input the text to be corrected into the text correction model, output the corresponding prediction tag for each token, and use the tag to correct the text. Due to time constraints, we modify the training set and validation set of the task using the font confusion dataset and phonetic confusion dataset as the training data of the text correction model.

The model architecture mainly consists of a MLM pre-trained model and an MLP (multi-layer perceptron). After the text is input into the pre-trained model, the token embeddings containing the relationship between the tokens and the semantic information of the tokens are obtained by calculating the interaction attention between tokens. Subsequently, all the token embeddings are input into the MLP structure, and the confidence distribution of each token regarding the label is obtained through a SoftMax function. The token with the highest score is taken as the final result to modify the original text.

3.2 Named Entity Recognition Module

Our system treats the NER task as a sequence labeling problem. The main purpose of the entity recognition module is to identify the span and type of entities in the text. The entity recognition module of the system is mainly composed of two types of network architectures, both of which use pre-trained models to enhance performance. In fact, we will explore different pre-trained models for each NER model structure.

3.2.1 Basic Pre-trained NER Network

This network structure, based on a pre-trained model, mainly consists of an embedding layer and a linear CRF(Lafferty et al., 2001) layer. Upon receiving a sentence as input, the embedding layer of the model encodes the sequence, and obtains the token representation of the input sentence. The token embedding is then fed into the CRF layer to obtain the conditional probability of the output. The inference of the model utilizes the Viterbi(Forney, 1973) algorithm to decode, and thus obtains the final sequence labeling result, as well as the entity span and the corresponding entity type.

3.2.2 Gazetteer NER Network

Our system uses the Gazetteer-Adapted Integration Network proposed by USTC-NELSLIP(Chen et al., 2022) to fuse the gazetteer information. In order to fuse the information of gazetteer, we need to build a prefix tree according to gazetteer in advance. When a sentence is input into the prefix tree, the longest matching algorithm is employed to generate a 73-dimensional one-hot vector. Unlike their two-stage training methods, we adopt the conventional one-stage training method to carry out supervised training by calculating the predicted results and the loss of actual tags.

3.3 Gazetteer

The key to building a gazetteer lies in the entity information stored in Wikidata. The entire Wikidata database can be viewed as a graph composed of numerous nodes. To classify these nodes, we employ a Graph Convolutional Network to obtain a probability distribution of the corresponding entity types. This method helps us to construct a gazetteer. We obtain all Chinese-related nodes and the inter-node relationships of Wikidata as a graph. Our system is based on a Graph Convolutional Network (GCN) classifier(Kipf and Welling, 2016), which is trained and validated on an existing labeling dataset to generate the classification results of each Chinese node in Wikidata.

In our system, the gazetteer plays two roles. Firstly, if the entities recognized by the entity module match those in the gazetteer, the type information of the corresponding entries in the gazetteer will be used as a feature by the stacking model to assess the correctness of the entities. Secondly, the gazetteer is also employed to enhance the performance of the entity recognition model, as illustrated in section 3.2.2.

3.4 Knowledge-based Classifier Module

The retrieval enhancement context has been proven to be effective for the named entity recognition task (Wang et al., 2021), wherein the candidate entity span is used as the input to retrieve the relevant text from the constructed knowledge system which is then fed into the text classification model. The main function of this module is to determine the corresponding type of the candidate entity span output by the entity recognition through the text classification method. Next, we will introduce how to establish a knowledge system and how to use knowledge to classify entity category.

Based on Wikipedia, we developed a local Wikipedia search engine to retrieve relevant information for input entities. We downloaded the latest version of Wikipedia and converted it into plain text, then indexed it using ElasticSearch (ES). ES is document-based and documents are the smallest search unit of ES. We used ES to search for candidate entities, obtaining descriptions related to them, as well as sentences containing them. Additionally, we leveraged the Google search engine to obtain entity-related information. After conducting comparative experiments, we chose Wikipedia as the data source for our knowledge system. The input for our classifier was composed of the entity, the original sentence in data released by MultiCoNER II and entity-related information, all concatenated together.

The information of the knowledge system will be used as the input of the text classification model, which will predict the category conditional probability of the entity. The classification model based on pre-training performs well in the current text classification task(Sinoara et al., 2019). Our text classification model is based on the pre-trained model and Multi-Layer Perceptron(MLP). Once a sentence is input into the model, the embedding layer encodes the sentence and outputs the corresponding token representation set. Following the commonly used method(Devlin et al., 2018), we feed the embedding corresponding to [CLS] from the token embedding representation set into the MLP layer for classification, input the output of the MLP layer into the SoftMax function, and output a probability distribution y about the classification label. Finally, the type label probability distribution of the candidate entity, y, is input into the stacking model in order to determine its correctness. For the experiment, multiple pre-trained models will be used and the best three will be selected. All the outputs of these three models are then input into the stacking model as features.

3.5 Stacking Model

In practice, the stacking approaches(Pavlyshenko, 2018) have achieved promising results in various tasks when it is used to fuse multiple features for classification. Our system outputs the probability distribution of multiple candidate entities and entity labels in the named entity recognition module, while in the gazetteer module and knowledge-based classifier, it obtains the probability characteristics of entity types. In this section, we will introduce the construction and training of the stacking model, which will be utilized to fuse the relevant features of the candidate entities and classify them in order to identify the correct entities from the candidate entity set as the system's final output.

The stacking model is a machine learning classifier used for supervised binary classification training. The stacking model is used to predict whether the entities in the candidate set are correct. It employs four different input features, namely NER Feature, Classification Feature, Gazetteer Feature and Entity Feature. NER Feature is the confidence level of the candidate entity output by the Entity Recognition Module, while Classification Feature is the probability distribution of the candidate entity label of the Knowledge-based Classifier. Gazetteer Feature is the label type and probability distribution of the candidate entity in the Gazetteer, and Entity Feature is the characteristics of the entity itself, such as its length. This model further utilizes the self-developed AutoML framework to automatically search for optimal models and parameters during parameter adjustment.

4 System Details

4.1 Five-Fold Cross-Validation

For each individual model of the named entity recognize module and the knowledge-based classifier module, the 5-fold cross-validation training strategy is adopted, and the outputs of the five models are finally fused. A 5-fold cross-validation training strategy is also applied in the evaluation. The prepared data are split into five pieces, each one is used as the validation set, while the other four pieces are used as the training set. After obtaining the five best models by this strategy, the logits of them are averaged to integrate them as an aggregated model. CRF models are just voted averagely.

4.2 Number of Nodes in Wikidata

The number of all nodes in Wikidata is too large. A large number of nodes need to be calculated in the Graph Convolutional Network. We need to delete some nodes to reduce the computational complexity. Wikidata contains entity nodes in all languages, especially leaf nodes, which account for the largest proportion, far exceeding the number of nodes we need. In the graph composed of Wikidata, we will delete all non-Chinese leaf nodes, greatly reducing the number of Wikidata nodes, and at the same time ensure that the graph structure features of Chinese leaf nodes are not damaged too much.

4.3 Model Selection

The experiment shows that different pre-trained models have a great impact on the effect of the experiment, and it is very important to evaluate multiple pre-trained models on the validation set. In the entity recognition module described in section 3.2, each entity recognition model structure will use multiple pre-trained models to test and select three best performing models. Similarly, for the text classification model described in section 3.4, we will also use multiple pre-trained models to test and select three best performing models. The experiment data is constructed based on the train set and validation set of Task 2. Finally, Basic Pre-trained NER network integrates Electra(Clark et al., 2020), Roberta(Liu et al., 2019) and Tacl-BERT(Su et al., 2021). We choose Electra, Roberta and CK-BERT(Zhang et al., 2022) for Gazetteer NER network. Nezha, Tacl-BERT and Roberta are integrated into Knowledge-based Classifier.

5 Results and Analysis

5.1 Main Results

In the challenge of SemEval 2023 Task 2: MultiCoNER II, our system only participated in the Chinese track and achieved the best results therein. Our noise subset and non-noise subset both had the highest F1 scores.

5.2 Effectiveness of Text Correction

There is few noise in the training set and validation set of the task, so text correction module is not needed to correct the text error. However, the data in the test set contains lots of text error, resulting in serious damage to the model performance. Table 1 is the F1 value on the test set before and after adding the text correction module. It can be seen that the text correction module improves the performance of entity recognition in multiple categories of data.

strategy	macro@F1
A	0.7336
B	0.7821
C	0.8288

Table 1: Results of the official test set on the Chinese track. Strategy A denotes our NER system without text correction module. Strategy B denotes our NER system with primary text correction module. Strategy C denotes our NER system with our best text correction module, which integrates more models and makes data augmentation.

5.3 Effectiveness of Each Module

In order to explore the effect of each module in the system, we used the released training set for training, and conducted experiments on the validation set. Due to the existence of overlapping samples in the training set and the validation set, in order to ensure the accuracy of the experimental results, the duplication was removed. Table 2 shows the effect of each module under different combinations. Because the validation set does not contain text noise, the text correction module is not added here. As shown in table, because the Knowledge-based Classifier Module and the Gazetteer Module integrate external knowledge, they significantly improve entity recognition performance of the system. Strategy D uses our self-developed AutoML technology, which greatly improves the performance of our system.

strategy	macro@F1
A	0.7817
B	0.8252
C	0.8676
D	0.9005

Table 2: Results of the official validation data on the Chinese track. Strategy A denotes NER system consisting of named entity recognition module and stacking model. Strategy B denotes NER system composed of named entity recognition module, knowledge-based classifier module and stacking model. Strategy C denotes NER system consisting of named entity recognition module, knowledge-based classifier module, gazetteer and stacking model. Strategy D denotes NER system consisting of named entity recognition module, knowledge-based classifier module, gazetteer and stacking model which trained by AutoML.

6 Conclusion

In this article, we present our system for the shared task, which won the championship in the Chinese track and far outperformed the second in the final score. To address the problem of text noise and complex entities out of knowledge base raised by the task, we designed a text correction module to correct text errors. We leveraged Graph Convolutional Network to construct a gazetteer with external knowledge to enhance the performance of the entity recognition model. Also, we employed a text classification model based on external knowledge to improve the accuracy of the entity type classification. Finally, we integrated multiple features into a stacking model to output the correct entities. Our experiments demonstrate that our NER system is able to address the complex entity recognition problem by constructing a gazetteer and utilizing knowledge retrieval. Additionally, we have demonstrated that building an effective text correction model can reduce the impact of text noise. The stacking model enables the integration of features of diverse dimensions, thereby effectively improving the performance of our entity recognition system. For future work, we intend to enhance the entity recognition performance by improving the accuracy of gazetteer construction and optimizing entity retrieval.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Sandeep Ashwini and Jinho D Choi. 2014. Targetable named entity recognition in social media. *arXiv preprint arXiv:1408.0782*.

Beiduo Chen, Jun-Yu Ma, Jiajun Qi, Wu Guo, Zhen-Hua Ling, and Quan Liu. 2022. Ustc-nelslip at semeval-2022 task 11: gazetteer-adapted integration network for multilingual complex named entity recognition. *arXiv preprint arXiv:2203.03216*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Leon Derczynski, Eric Nichols, Marieke Van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. MultiCoNER v2: a Large Multilingual dataset for Fine-grained and Noisy Named Entity Recognition.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. Gazetteer Enhanced Named Entity Recognition for Code-Mixed Web Queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681.

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.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Dou Hu and Lingwei Wei. 2020. Slk-ner: exploiting second-order lexicon knowledge for chinese ner. *arXiv preprint arXiv:2007.08416*.

Suzana Ilić, Edison Marrese-Taylor, Jorge A Balazs, and Yutaka Matsuo. 2018. Deep contextualized word representations for detecting sarcasm and irony. *arXiv preprint arXiv:1809.09795*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jouni Luoma and Sampo Pyysalo. 2020. Exploring cross-sentence contexts for named entity recognition with bert. *arXiv preprint arXiv:2006.01563*.

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.

Bohdan Pavlyshenko. 2018. Using stacking approaches for machine learning models. In *2018 IEEE second international conference on data stream mining & processing (DSMP)*, pages 255–258. IEEE.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Tjong Kim Sang and F Erik. 2002. Introduction to the conll-2002 shared task: language-independent named entity recognition. In *Proceedings of CoNLL-2002/Roth, Dan [edit.]*, pages 155–158.

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A study of the importance of external knowledge in the named entity recognition task. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 241–246.

Roberta A Sinoara, Jose Camacho-Collados, Rafael G Rossi, Roberto Navigli, and Solange O Rezende. 2019. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163:955–971.

Jana Straková, Milan Straka, and Jan Hajič. 2019. Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*.

Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier. 2021. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

Beth M Sundheim. 1995. Named entity task definition, version 2.1. In *Proc. Sixth Message Understanding Conf.(MUC-6)*, pages 317–332.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. Wikineural: Combined neural and knowledge-based silver data creation for multilingual ner. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Improving named entity recognition by external context retrieving and cooperative learning. *arXiv preprint arXiv:2105.03654*.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, et al. 2022. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. *arXiv preprint arXiv:2203.00545*.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. *arXiv preprint arXiv:2005.07150*.

Taolin Zhang, Junwei DOng, Jianing Wang, Chengyu Wang, Ang Wang, Yinghui Liu, Jun Huang, Yong Li, and Xiaofeng He. 2022. Revisiting and advancing chinese natural language understanding with accelerated heterogeneous knowledge pre-training. *arXiv preprint arXiv:2210.05287*.

A Training Data

Our system consists of multiple distinct modules, each with different training data. The text correction module utilizes our own constructed and annotated text correction dataset in the news field, in addition to font confusion and phonetic confusion datasets. The font confusion dataset is utilized to test and evaluate text correction systems, comprising pairs of characters from different fonts that are visually similar and easily confused. The phonetic confusion dataset is employed to evaluate text correction systems too, consisting of a group of phonetically similar words from multiple languages that are easily confused. For example, in English, "cat" and "cut" sound very similar but have completely different meanings. We further enhance the data using the font and phonetic confusion datasets on top of the news field text correction dataset.

The construction of GCN's training data is based on the training and validation sets released with the MultiCoNER II, as described in section 3.3. Entities and their relationships in Wikidata are abstracted into a graph model, where GCN can predict the class of an entity based on its relevant features in the graph, which includes entities present in both the training and validation sets.

The training data for the remaining modules is entirely based on the training and validation sets released by MultiCoNER II.

B Stacking Model and AutoML

The feature set for the stacking model has been described in section 3.5. The purpose of the stacking model is to integrate the outputs of various modules to obtain a more reasonable result. In our system, most modules use multiple models, and compared to a simple voting method, the stacking approach is a more reasonable and effective choice. Previous research and competition solutions have also demonstrated that the stacking method can effectively improve performance.

The AutoML framework we have developed is capable of performing automated operations such as data cleansing, missing value imputation, feature selection, and transformation to enhance the quality and usability of data. It also has the capability of model selection and optimization, automatically selecting the optimal algorithm and hyperparameter combination to improve model performance. Furthermore, it can perform model training and evaluation, automatically conducting model training and cross-validation to assess the accuracy and generalization ability of the model. In the process of constructing a stacking model, we employ our self-developed AutoML framework to assist us with model selection and hyperparameter optimization, thus achieving optimal results.

C Detail Relevant to Text Correction Module.

The input of the text correction module comprises erroneous text, and its output generates a label for each token in the text. There are two types of labels: replacement and correct. If a token in the original text corresponds to the correct label, it implies that the text does not require any modification. On the other hand, if a token in the original text corresponds to the replacement label, it implies that the text requires modification to the correct word. The number of replacement label categories is equivalent to the number of entries in the vocabulary. If the replacement label is "delete|x", it suggests that the corresponding text token should be replaced with x.

In Table 1 of section 5.2, strategy B implemented the primary text correction model, which relied solely on the results trained using the news correction dataset. Strategy C employed data augmentation through the use of the font confusion dataset and the phonetic confusion dataset, both of which are high-quality datasets that have been manually annotated, thereby enhancing the performance of the text correction module.

D Relevant Pre-trained Models for System.

As a matter of fact, section 4.3 has already provided an introduction to the pre-trained language models employed in the named entity recognition module and the knowledge-based classifier module. In the text correction module, we have utilized the pre-trained model, Roberta(Liu et al., 2019).