WkNER: Enhancing Named Entity Recognition with Word Segmentation Constraints and kNN Retrieval

Yanchun Li[1,2,3], **Senlin Deng**[1], **Dongsu Shen**[1,2,3]*, **Shujuan Tian**[1,2,3]
Saiqin Long[4]
[1]School of Computer Science & School of Cyberspace Science , Xiangtan University
[2]Hunan International Scientific and Technological Cooperation Base of Intelligent Network
[3]Key Laboratory of Hunan Province for Internet of Things and Information Security
[4]College of Information Science and Technology, Jinan University
{ycli, dsshen, sjtianwork}@xtu.edu.cn
dengsenlin6150@gmail.com, saiqinlong@jnu.edu.cn

Abstract

Fine-tuning Pre-trained Language Models (PLMs) is a popular Natural Language Processing (NLP) paradigm for addressing Named Entity Recognition (NER) tasks. However, neural network models often demonstrate poor generalization capabilities due to significant disparities between the knowledge learned by PLMs and the distribution of the target dataset, as well as data scarcity issues. In addition, token omission in predictions due to insufficient learning remains a challenge in NER. In this paper, we propose a kNN retrieval enhancement algorithm (WkNER) that incorporates word segmentation information to enhance the model's generalization ability and alleviate the problem of missing entity tokens in prediction. The introduction of word segmentation information is used to preliminarily determine the boundaries of entities and alleviate the common prediction errors of missing tokens within entities made by the fine-tuned model. Secondly, we find that non-entities in the retrieval table contain a large amount of redundant information, and explore the effects of introducing non-entity information of different scales on the model. Experimental results show that our proposed method significantly improves the performance of baseline models, and achieves better or compared recognition accuracy than previous state-of-the-art models in multiple public Chinese and English datasets. Especially in low-resource scenarios, our method achieves higher accuracy on 20% of the dataset than the original method on the full dataset.

Keywords: Named Entity Recognition, Knowledge Enhancement, Low-resource Scenarios

1. Introduction

Named Entity Recognition (NER) aims to identify and locate entities in text, and is one of the hot research tasks in Natural Language Processing (NLP). NER can be used to handle structured and unstructured data, and is an important foundational tool for many advanced semantic analysis tasks, such as Relation Extraction, Knowledge Graphs, and more. NER requires detecting the span and category of the entity from the text block, and only when the detected boundaries and categories are consistent with the label can the entity be considered correctly identified.

Most existing works formalize the NER task as a sequence labeling problem (Knight et al., 2016; Tang et al., 2018; Straková et al., 2019a), where each token is assigned a specific label to indicate its belonging to a certain entity category (such as person names, locations, organization names, etc.). However, due to the presence of boundary words like articles or labeling errors, the boundaries of entities can be ambiguous, making the judgment of boundaries prone to confusion in sequence labeling methods. Additionally, in the training scenarios of low-resource or long-tail datasets, it is common for the model to have insufficient learning for certain entity categories, which easily leads to the model's missing predictions for tokens in that category. As shown in Figure 1, (a) represents a case where one

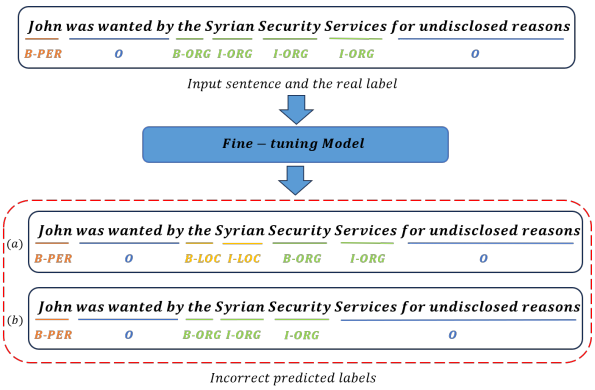

Figure 1: Two types of common errors in entity prediction in sequence labeling models are illustrated. (a) represents inaccurate entity type recognition, and (b) represents inaccurate boundary delineation of entities.

* Corresponding authors

LREC-COLING 2024, pages 17651–17663

entity is mistakenly identified as two different entity types; (b) demonstrates inaccurate recognition of entity boundaries, where tokens that should be recognized as 'I-ORG' are wrongly identified as 'O', resulting in missing token predictions. Therefore, some works suggest that incorrect boundaries are the main reason of entity recognition errors (Wang et al., 2019; Eberts and Ulges, 2020).

Conditional Random Fields (CRF) (Collobert et al., 2011) is a probabilistic graphical model used for sequence labeling and structured prediction, often used to enforce label transitions. CRF improves the modeling and prediction capabilities of structured data by considering the contextual relationships in the input data and label dependencies. Typically, constraint rules of CRF are set, such as: assuming that if a token is predicted as 'B-ORG' type, then the next token is likely to be 'I-ORG', and the first token of an entity cannot start with 'I-', etc., to enhance the accuracy of the model. Although the two error scenarios in Figure 1 satisfy the label dependency of the CRF mentioned above, they do not conform to the true labels. We believe this is due to the model having too little exposure to such entity samples during the learning phase, and the training of the CRF introduces more training parameters, resulting in poor generalization of the model towards the boundaries of entities. This paper proposes an initial segmentation of the input sentence to obtain word segmentation information that can effectively constrain entity boundaries and solve these two common issues.

In addition, when facing long-tailed datasets or limited samples, the performance of sequence labeling models is often unsatisfactory and prone to overfitting (Wang et al., 2022; Das et al., 2022). Existing research has found that kNN can achieve excellent unconditional language modeling during the inference stage (Khandelwal et al., 2020; He and Choi, 2021). According to the definition given by Hastie et al. (2009), kNN is a lazy learner, which can avoid overfitting of parameters and effectively smooth the influence of isolated noisy training data (Boiman et al., 2008). The kNN-NER (Wang et al., 2022) mitigates the long-tail problem to some extent by introducing kNN retrieval enhancement, but requires a large amount of computational resources during the retrieval process due to the use of both entity and non-entity information. In fact, in a dataset, the non-entity information accounts for a significant proportion, while the entity information in a dataset is very sparse. Therefore, introducing excessive non-entity information does not bring gain, but rather affects the enhancement effect of the basic model to a certain extent. Secondly, kNN-NER does not make full use of prior knowledge in the dataset and fails to solve the problem of entity boundary ambiguity shown in Figure 1.

In this paper, we propose WkNER to address the issues mentioned above, which combines word segmentation information with the kNN retrieval enhancement algorithm to provide a label probability distribution containing entity word segmentation boundary information for the basic model. We use existing word segmentation tools to extract entity word segmentation information, which can help the fine-tuned model initially determine the boundaries of entities, thus mitigating the common issue of missing token predictions for entities in the basic model. Additionally, regarding the issue of significant resource consumption caused by the introduction of all non-entities, we believe that non-entity information should be appropriately pruned to reduce the interference of redundant non-entity information. Our method can offer more efficient data distribution information to the basic model, allowing it to achieve equivalent results without training on the complete dataset, which further enhances the model's performance in low-resource scenarios. To validate the superiority of our WkNER, we conduct a series of experiments on widely used baseline models and datasets, and the experimental results demonstrate that our method achieves optimal performance on multiple quantitative indicators.

In summary, our contributions are as follows:

- We propose WkNER, which introduces word segmentation information during the inference phase, providing the model with entity boundary information, enhancing the basic model's ability to predict entity boundaries, and improving generalization performance under low-resource scenarios.

- We explore the construction of retrieval tables using non-entity information of different scales to effectively utilize entity information in the dataset. The experiments have shown that by introducing an appropriate amount of non-entity information, the performance of retrieval enhancement algorithms can be significantly improved, which proves that the presence of excessive non-entity information in the dataset is redundant.

- Our method significantly improves the performance of the baseline model on widely used Chinese and English datasets, achieving state-of-the-art results on the Chinese OntoNotes 4.0 and MSRA datasets. Importantly, our algorithm can enhance the performance of the model trained on a 20% training set to the level equivalent to the model trained on the complete dataset.

2. Related Work

2.1. Named Entity Recognition

Research on NER tasks in the field of NLP has a long history and is a fundamental task in information extraction. The tasks in this field are mainly divided into flat NER tasks and nested NER tasks, which are usually solved by three types of methods: token-based methods, span-based methods, and generation-based methods. The token-based approaches typically assign a maximum confidence label to each token. Hammerton (2003) were the first to use a sequence tagging model for NER tasks, in which they attempted to use a unidirectional Long Short-Term Memory Network (LSTM) to solve the task. Collobert et al. (2011) introduced CRF into neural network-based sequence tagging models, enabling explicit encoding of the transition possibilities between adjacent labels. The use of PLMs (e.g., ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019))) for feature extraction has been widely accepted and has further improved the performance of NER. Token-based methods are applicable for flat NER tasks. For nested NER tasks, existing works suggest using span-based and generation-based methods as solutions. For the span-based methods, Li et al. (2020b) transformed NER tasks into a Machine Reading Comprehension (MRC) problem, which provides a unified solution for both nested NER and flat NER tasks, achieving good results. Zheng et al. (2019) and Shen et al. (2021b) proposed span classification with boundary detection and boundary regression strategies, respectively, to address nested NER tasks. Generation-based methods (Yan et al., 2021; Lu et al., 2022; Zhang et al., 2022) are also commonly used to tackle both flat NER and nested NER tasks, by generating text containing recognized entities and label sequences as the detection results. However, the exhaustive search in span-based methods (Li et al., 2020b) and the generation processing in generation-based methods (Yan et al., 2021; Lu et al., 2022; Zhang et al., 2022) require a large amount of sample resources and have significant time consumption during the inference stage. Therefore, in this article, we mainly focus on flat NER tasks in low-resource scenarios, leaving nested NER tasks for our future work.

2.2. External Knowledge Enhancement

Neural network models often face issues such as data scarcity and poor feature extraction during the training process, which can result in decreased generalization performance when encountering rare entities. To enhance the model's generalization performance in low-resource scenarios, incorporating external information through retrieval or fusion can provide valuable prior knowledge. Sun et al. (2021) introduced the visual and phonetic information of text into PLMs, integrating these details in the word embedding stage. This approach effectively addresses ambiguity and enriches the representation of word embeddings. On the other hand, Rei (2017) added lexical feature information to the NER task model and improved the handling of entity boundaries, resulting in a significant improvement in model performance. Character-level representations of lexical features are commonly used for English tasks (Huang et al., 2015; Knight et al., 2016), while lexical information is helpful in the Chinese NER domain (Ma et al., 2020; Li et al., 2020a). Previous research has also employed other external sources of information as features to formulate hybrid representations, such as morphological features (Xu et al., 2019) and toponyms of named entities (Gu et al., 2018). Moreover, leveraging the distributional information of training data through retrieval has become a popular approach in recent years. Wang et al. (2022) and Khandelwal et al. (2020) enhanced the performance of models in low-resource scenarios by utilizing the distributional information of training data. They combined kNN with sequence labeling models and used the probability distribution obtained from kNN retrieval to enhance the performance of the fine-tuned model.

3. Methodology

In this section, we provide detailed descriptions of the backbone and our proposed algorithm. In section 3.1, we introduce the basic model architecture, which is utilized as a text feature extractor in our algorithm. In section 3.2, we describe the overall process of our algorithm, including the construction of the retrieval sets, the introduction of word segmentation information in the inference stage, and the retrieval enhancement of the fine-tuned model.

3.1. Fine-tuning Language Model

We treat NER as a sequence labeling task, use a fine-tuned model to convert tokens into high-dimensional vectors, and then obtain the probability distribution of tokens on the labels through a $Softmax$ layer. The process can be formalized as follows: for an input sentence $\boldsymbol{x}=\{x_1, x_2, \cdots, x_{N-1}, x_N\}$ with a length of N, where $x_i\,(1 \leq i \leq N)$ represents the i-th token in the sentence. Therefore, all the tokens in the training set can be defined as $X=\left\{x_i^k | 1 \leq k \leq K, 1 \leq i \leq N_k\right\}$, where x_i^k represents the i-th token in the k-th training sample. K is the number of samples in the training set, and N_k is the length of the k-th sample. To classify each token in the input sentence into M labels, the M

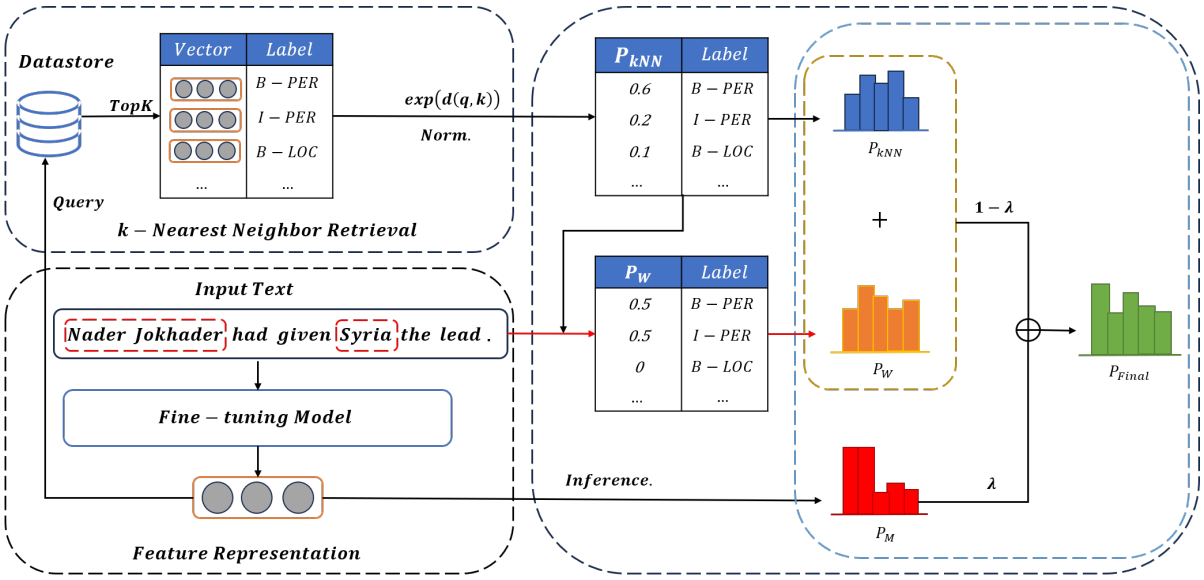

Figure 2: Illustration of the overall process of WkNER. The part of Feature Representation describes the extraction of text feature vectors and word segmentation information by the fine-tuned model. The part of k-Nearest Neighbor Retrieval describes the retrieval enhancement using kNN to obtain the probability distribution of P_{kNN}. The right part describes the enhancement of the probability distribution P_M of the fine-tuned model using probability distributions P_{kNN} and P_W. P_W is obtained by combining P_{kNN} with word segmentation information to assign a word segmentation probability distribution P_W for each token in the word segmentation boundary.

labels can be represented as $Y=\{y_j|1 \leq j \leq M\}$, where y_j represents the j-th category. The task of sequence labeling is to assign a category y_j to each token x_i in $\boldsymbol{x}$, which is a multi-classification task for each token x_i. We use PLMs (such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)) as an encoder to extract features h_i from the i-th token x_i in the sentence, where $h_i \in \mathbb{R}^D$ and D represents the dimension of the vector space, then h_i can be represented as follows:

$$h_i = f(x_i, \boldsymbol{x}) \tag{1}$$

where $f(\cdot,\cdot)$ is the fine-tuned PLM; h_i is the feature vector of token x_i in the context $\boldsymbol{x}$, and the acquisition of the query vector in the subsequent kNN algorithm also follows the above equation.

Therefore, the fine-tuned model predicts the probability distribution of token x_i on each label as follows:

$$P_M(y_j|x_i, \boldsymbol{x}) = Softmax(MLP(h_i)) \tag{2}$$

where x_i is the token in the input sentence $\boldsymbol{x}$, and MLP refers to the multi-layer perceptron connected after the PLMs.

3.2. WkNER

The overall process of WkNER is shown in Figure 2. To articulate the algorithmic idea introduced by the word segmentation information more clearly, we first introduce the construction of the kNN retrieval table in section 3.2.1. Then, section 3.2.2 describes the extraction of word boundary information and the computation process of the probability distribution P_W of word segmentation information. Finally, section 3.2.3 presents how to enhance the fine-tuned model using the distribution of word segmentation information P_W and the data augmentation distribution P_{kNN}.

3.2.1. Building Retrieval Table

The retrieval table stores data in the form of key-value pairs (k_i, v_i). Here, the key k_i refers to the token vector h_i of each token in the training dataset, and the corresponding value v_i represents the entity type y_j of that token. The retrieval table can be represented mathematically as follows:

$$\begin{aligned} D &= (K, V) \\ &= \{(f(x_i, \boldsymbol{x}), y_j)|x_i \in X, y_j \in Y\} \end{aligned} \tag{3}$$

where $f(x_i, \boldsymbol{x})$ is equivalent to h_i in Formula 1.

3.2.2. Enhancing with Word Segmentation Information

To extract word segmentation information, we use the existing word segmentation tool Hanlp (He and Choi, 2021) to segment the input sentence $\boldsymbol{x}=\{x_1, x_2, \cdots, x_{N-1}, x_N\}$. After

segmentation, the sentence can be represented as $\boldsymbol{x}=\{w_1, w_2, \cdots, w_{K-1}, w_K\}$, where $w_k=\{x_{k,1}, x_{k,2}, \cdots, x_{k,S-1}, x_{k,S}\}\,(1 \leq k \leq K)$ represents the k-th chunk in the input sentence $\boldsymbol{x}$, consisting of a sequence of S consecutive tokens from $\boldsymbol{x}$, which can be a noun phrase or a combination sequence of numbers, such as birthday. Detailed segmentation algorithm can be found in Appendix A.1. By utilizing the probability distribution obtained from kNN, we initially classify the tokens within each word segmentation. Thus, the set of category labels for the tokens in w_k can be represented as T_k:

$$T_k = \Big\{ \arg\max_{y_j} P_{kNN}\left(y_j \mid x_{k,i}, \boldsymbol{x}\right) \mid 1 \leq i \leq S_k \Big\} \tag{4a}$$

$$L = g\left(T_k\right) \tag{4b}$$

where S_k is the length of the k-th chunk. The function $g\left(\cdot\right)$ represents obtaining the element with the highest count in a set. Therefore, L is the most frequent label in the set T_k. The probability distribution of chunk w_k across various labels can be represented as follows:

$$P_W\left(y_j \mid w_k, \boldsymbol{x}\right) = \begin{cases} 1, & y_j = L \\ 0, & y_j \neq L \end{cases} \tag{5}$$

where $P_W\left(y_j \mid w_k, \boldsymbol{x}\right)$ is the initial word segmentation information, indicating the probability distribution of possible labels for the chunk w_k. Further processing is required to normalize this probability distribution and expand it to each token $x_{k,i}$ within the chunk w_k. Mathematically, this can be expressed as:

$$P_W\left(y_j \mid x_{k,i}, \boldsymbol{x}\right) = P_W\left(y_j \mid w_k, \boldsymbol{x}\right), 1 \leq i \leq S_k \tag{6}$$

where $P_W\left(y_j \mid x_{k,i}, \boldsymbol{x}\right)$ is the final word segmentation boundary information for each token in w_k.

3.2.3. Retrieval Enhancement for Fine-tuned Model

During the inference stage of the model, for each token x_i in the input sentence $\boldsymbol{x}$, the corresponding token vector $f(x_i, \boldsymbol{x})$ is generated through the fine-tuned PLM. This vector $f(x_i, \boldsymbol{x})$ is then used to retrieve the k elements with the highest similarity from the retrieval table D. The similarity $d\left(\cdot,\cdot\right)$ in this context is measured with the L^2 norm. To improve the retrieval speed of the model, we utilize the third-party search library Faiss[1] (Johnson et al., 2021) to construct a nearest neighbor retrieval table for efficient searching of the $TopK$ similar records. Then, the similarity between the query vector $f(x_i, \boldsymbol{x})$ and the retrieval elements is mapped to the probability distribution of the corresponding labels:

$$P_{kNN}\left(y_j = val \mid x_i, \boldsymbol{x}\right) \propto \sum_{(key,val)\in D} \mathbb{I}_{y_j = val} exp\left(-\frac{d\left(f(x_i, \boldsymbol{x}), key\right)}{\tau}\right) \tag{7}$$

where τ is a hyperparameter used to control the level of smoothness of the probability distribution.

Integrating word segmentation boundary information and kNN retrieval probability distribution information to adjust the probability distribution of the fine-tuned model, the final probability distribution P_{Final} is gained.

$$P_{Final}(y_j \mid x_i, \boldsymbol{x}) = \lambda P_M(y_j \mid x_i, \boldsymbol{x}) + (1-\lambda)[P_{kNN}(y_j \mid x_i, \boldsymbol{x}) + P_W(y_j \mid x_i, \boldsymbol{x})] \tag{8}$$

where $\lambda \in [0, 1]$ is a hyperparameter that balances the enhanced probability distribution with the predicted distribution of the fine-tuned model.

The specific algorithm workflow is as follows:

Algorithm 1 Enhancement of Word Boundary Information

Input: $\boldsymbol{x}=\{x_1, x_2, ..., x_{N-1}, x_N\}$, $P_{kNN}(y_j \mid x_i, \boldsymbol{x})$ and λ;
Output: $P_{Final}(y_j \mid x_i, \boldsymbol{x})$;

$P_M(y_j \mid x_i, \boldsymbol{x}) = Softmax(MLP(f(x_i, \boldsymbol{x})))$;
Segment the input sentence $\boldsymbol{x}$ into $\boldsymbol{x}=\{w_1, w_2, ..., w_{K-1}, w_K\}$ using the word segmentation tool.
for all $k = 1, 2, \cdots, K$ **do**
  $l = startIndex(w_k)$;
  $r = endIndex(w_k)$;
  $label\_count[0 : M+1] = \{0\}$;
  for all $k = l, l+1, \cdots, r-1, r$ **do**
    $y = \arg\max_{y_j} P_{kNN}(y_j \mid x_{k,i}, \boldsymbol{x})$;
    Increment $label\_count[y]$.
  end for
  $label = \arg\max_h \ label\_count[h]$;
  for all $j = 1, 2, \cdots, M$ **do**
    if $label = y_j$ **then**
      $P_W(y_j \mid w_k, \boldsymbol{x}) = 1$;
    else
      $P_W(y_j \mid w_k, \boldsymbol{x}) = 0$;
    end if
    for all $i = 1, 2, \cdots, S_k$ **do**
      $P_W(y_j \mid x_i, \boldsymbol{x}) = P_W(y_j \mid w_k, \boldsymbol{x})$;
    end for
  end for
end for
$P_{Final}(y_j \mid x_i, \boldsymbol{x}) = \lambda P_M(y_j \mid x_i, \boldsymbol{x}) + (1-\lambda)\,[P_{kNN}(y_j \mid x_i, \boldsymbol{x}) + P_W(y_j \mid x_i, \boldsymbol{x})]$;
return $P_{Final}(y_j \mid x_i, \boldsymbol{x})$;

[1] For more detailed usage instructions, please refer to https://github.com/facebookresearch/faiss/wiki/Indexing-1G-vectors

Model	CoNLL 2003			Ononotes 5.0		
	Pr.	**Re.**	**F1**	**Pr.**	**Re.**	**F1**
Base Model						
BERT-Base (Devlin et al., 2019)	91.10	91.02	91.06	85.77	86.30	86.03
+kNN-NER (Wang et al., 2022)	91.50	91.58	91.54	85.89	86.49	86.19
+WkNER (our)	93.08	91.27	**92.17**	86.09	86.35	**86.22**
Large Model						
BERT-Large (Devlin et al., 2019)	91.89	92.67	92.28	86.47	87.81	87.14
+kNN-NER (Wang et al., 2022)	92.26	92.43	92.40	86.49	88.10	87.29
+WkNER (our)	93.96	92.87	**93.41**	86.83	87.83	**87.33**
RoBERTa-Large (Liu et al., 2019)	91.12	91.82	91.47	86.68	87.98	87.32
+kNN-NER (Wang et al., 2022)	91.20	91.85	91.52	86.73	88.29	87.51
+WkNER (our)	93.44	91.87	**92.65**	87.16	87.98	**87.57**

Table 1: Comparison on English datasets: OntoNotes 5.0 and CoNLL 2003.

Model	OntoNotes 4.0			MSRA			Weibo NER		
	Pr.	**Re.**	**F1**	**Pr.**	**Re.**	**F1**	**Pr.**	**Re.**	**F1.**
Base Model									
BERT-Base (Devlin et al., 2019)	78.32	82.27	80.25	94.95	94.64	94.79	67.21	69.81	68.48
+kNN-NER (Wang et al., 2022)	80.23	81.60	80.91	95.34	94.64	94.99	68.37	71.01	69.67
+WkNER (our)	81.14	82.81	**81.97**	95.77	95.32	**95.54**	71.71	69.81	**70.75**
RoBERTa-Base (Liu et al., 2019)	78.59	82.39	80.44	95.12	95.10	95.11	67.12	71.01	69.01
+kNN-NER (Wang et al., 2022)	78.67	82.73	80.65	95.61	94.93	95.27	67.97	71.26	69.58
+WkNER (our)	81.74	83.60	**82.66**	96.19	95.61	**95.90**	71.22	70.53	**70.87**
ChineseBERT-Base (Sun et al., 2021)	80.06	83.33	81.66	95.31	95.46	95.39	69.17	68.84	69.01
+kNN-NER (Wang et al., 2022)	81.43	82.58	82.00	95.73	95.27	95.50	68.97	73.71	**71.26**
+WkNER (our)	81.54	84.11	**82.81**	95.97	95.84	**95.90**	72.11	69.32	70.69
Large Model									
RoBERTa-Large (Liu et al., 2019)	80.69	82.56	81.62	95.46	95.53	95.50	69.62	70.29	69.95
+kNN-NER (Wang et al., 2022)	80.60	82.78	81.68	96.16	95.07	95.61	69.63	71.98	70.78
+WkNER (our)	82.46	83.52	**82.99**	96.64	96.16	**96.40**	72.48	71.26	**71.86**
ChineseBERT-Large (Sun et al., 2021)	81.17	83.32	82.23	95.86	95.32	95.59	67.18	73.67	70.28
+kNN-NER (Wang et al., 2022)	80.75	84.98	82.81	95.83	95.68	95.76	68.69	74.64	71.53
+WkNER (our)	83.06	84.57	**83.81**	96.58	95.67	**96.12**	70.44	74.10	**72.22**

Table 2: Comparison on Chinese datasets: OntoNotes 4.0, MSRA, and Weibo NER.

4. Experiments

4.1. Datasets and Metrics

We use multiple publicly available Chinese and English datasets to evaluate our method. The Chinese datasets include: OntoNotes 4.0 (Pradhan et al., 2011), MSRA (Levow, 2006), and Chinese Weibo NER (Peng and Dredze, 2015); and the English datasets include: CoNLL 2003 (Sang and Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2013). The above-mentioned datasets are all for the flat NER tasks, and our evaluation criteria are based on entity-level Precision, Recall, and micro F1-score. The detailed information of the dataset can be found in Appendix A.2.

4.2. Backbone Models

In order to evaluate whether the gain effect of WkNER is effective on different backbone models. Similar to Wang et al. (2022), on the English dataset, we use the base (768 hidden size, 12 layers) and large (1024 hidden size, 24 layers) size of BERT (Devlin et al., 2019), and the large size of RoBERTa (Liu et al., 2019). On the Chinese dataset, we use the base size of BERT, the base and large size of RoBERTa, and ChineseBERT (Sun et al., 2021). ChineseBERT is an improved model that integrates phonetic and character information, enhancing the model's ability to model Chinese language corpus better. The settings of the hyperparameters involved in the experiment can be found in Appendix A.3

4.3. Results and Analysis

4.3.1. Comparison on the Complete Datasets

Tables 1 and 2 present the experimental results on the complete Chinese dataset and English dataset respectively. On each dataset, we compare the performance of our algorithm with the kNN-NER and the baseline models on different PLMs. Specifically, on the Chinese OntoNotes 4.0 dataset, based on the RoBERTa PLM, our algorithm achieves a significant improvement of +2.22% in F1-score compared to the baseline model, and a notable im-

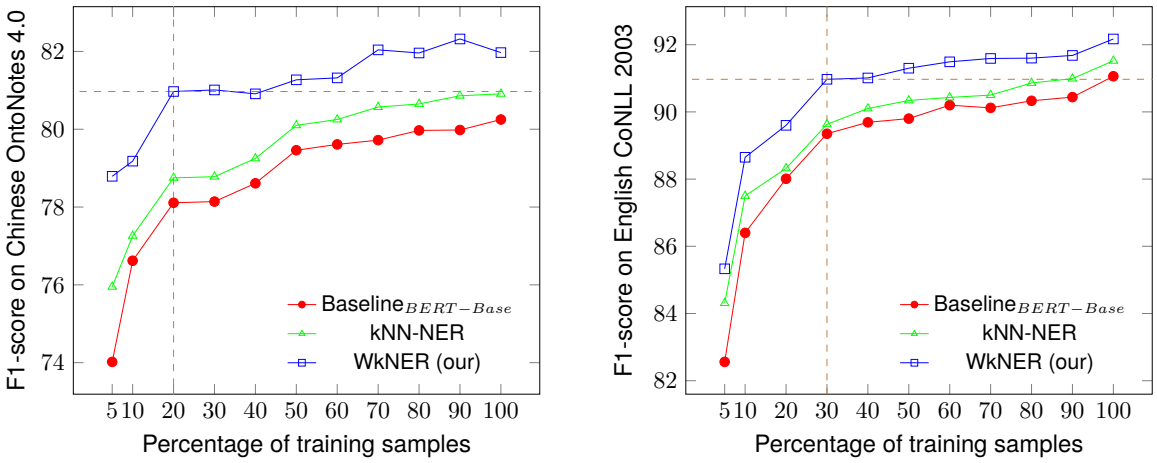

Figure 3: Experimental results under low-resource settings on the Chinese OntoNotes 4.0 and English CoNLL 2003 datasets.

Model	CoNLL 2003		
	Pr.	**Re.**	**F1**
Devlin et al. (2019)	–	–	92.80
Li et al. (2020b)	92.33	**94.61**	93.04
Yu et al. (2020)	93.70	93.30	93.50
Shen et al. (2021a)	92.13	93.73	92.94
Zhu and Li (2022)	93.61	93.68	**93.65**
Shen et al. (2023)	92.96	93.18	93.08
WkNER$_{BERT}$	**93.96**	92.87	93.41
Model	OntoNotes 4.0		
	Pr.	**Re.**	**F1**
Ma et al. (2020)	**83.41**	82.21	82.81
Li et al. (2020b)	82.98	81.25	82.11
Chen and Kong (2021)	79.25	80.66	79.95
Zhu and Li (2022)	81.65	84.03	82.83
WkNER$_{ChineseBERT}$	83.06	**84.57**	**83.81**
Model	MSRA		
	Pr.	**Re.**	**F1**
Ma et al. (2020)	95.75	95.10	95.42
Li et al. (2020b)	96.18	95.12	95.75
Wu et al. (2021)	–	–	96.24
Zhu and Li (2022)	96.37	96.15	96.26
WkNER$_{RoBERTa}$	**96.64**	**96.16**	**96.40**
Model	Weibo NER		
	Pr.	**Re.**	**F1**
Ma et al. (2020)	–	–	70.50
Li et al. (2020b)	–	–	68.55
Shen et al. (2021a)	70.11	68.12	69.16
Chen and Kong (2021)	–	–	70.14
Wu et al. (2021)	–	–	70.43
Zhu and Li (2022)	70.16	**75.36**	**72.66**
WkNER$_{ChineseBERT}$	**70.44**	74.10	72.22

Table 3: Results of Named Entity Recognition on Chinese and English datasets.

provement of +2.01% compared to the kNN-NER. Table 3 shows the comparison results between our method and the previous state-of-the-art methods. It is worth noting that on the Chinese OntoNotes 4.0 and MSRA datasets, compared to the previous state-of-the-art methods, our algorithm achieves F1-score improvements of +0.98% and +0.14%, respectively.

From Tables 1 and 2, it can be seen that introducing segmentation information during the retrieval process has a much better enhancement effect on the baseline models than that of the kNN-NER, proving the effectiveness of our proposed method in improving model performance. This is because the extracted word segmentation information provides the baseline models with more prior knowledge and allows the models to better correct entity boundaries, thereby addressing the issue of missing entity tokens in predictions. In addition, we find that the improvement effect of the enhancement algorithm on the baseline model is not very obvious in the Chinese MSRA and the English OntoNotes 5.0 datasets. This is because the information on various entity types in the Chinese MSRA and English OntoNotes 5.0 datasets is abundant and balanced, allowing the fine-tuned model to learn various entity information well during the training process. In this case, the model itself has a strong generalization ability for entity boundary prediction, so it does not rely heavily on the prior knowledge of the enhanced model. However, in other datasets with less training data, due to the extremely imbalanced entity types and overall smaller dataset scale, the model is prone to overfitting and other issues, resulting in poor prediction of entity boundaries. Therefore, the enhancement algorithm needs to provide more prior knowledge to improve performance, which is also the reason why our algorithm performs well on these datasets. This proves that our algorithm not only improves the performance of the baseline model on general datasets but also performs better in low-resource scenarios.

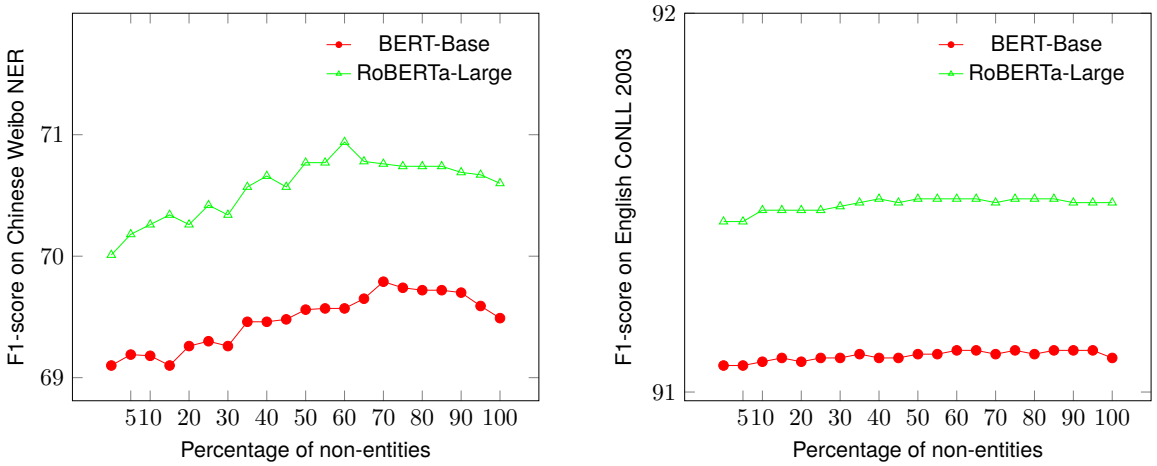

Figure 4: Ablation studies of non-entity information at different scales. The effect of utilizing non-entity information of different scales on kNN-NER. The experimental parameter is set as k=128 and evaluated under three different random seeds, with the average taken as the result.

4.3.2. Comparison under Low-resource Scenarios

Figure 3 demonstrates the performance of our algorithm, kNN-NER, and the baseline model on training sets of different scales. On the Chinese OntoNotes 4.0 dataset, our algorithm achieves a performance improvement of +4.77% on the baseline model by fine-tuning the BERT PLM with only 5% of the data, which is a +2.77% higher improvement compared to kNN-NER. Moreover, our algorithm surpasses the performance of the model fine-tuned on the full dataset when only 20% of the data is used. On datasets of other scales, the enhancement effect of our algorithm outperforms kNN-NER's enhancement effect on the baseline models. This demonstrates the significance of the word segmentation information extracted by our algorithm as a crucial form of prior knowledge, greatly improving the fine-tuned model's generalization performance. It also reflects that models fine-tuned on low-resource datasets are prone to errors in missing entity tokens during the entity prediction process, as shown in Figure 1. This issue is caused by the sparsity of entities in the data, as the lack of data hinders the model from effectively learning the distribution of different entity categories. Therefore, it is necessary to introduce word segmentation information into the fine-tuned model through the enhancement algorithm to improve its performance.

4.4. Ablation Study

We perform ablation studies on the Chinese OntoNotes 4.0, Weibo NER and English CoNLL 2003 datasets to separately analyze the effects of introducing word segmentation information in WkNER and using non-entity information of different scales within WkNER.

4.4.1. Word Segmentation Information

In this section, we investigate the effect of adding word segmentation information to kNN-NER. From Table 4, we can observe that with the introduction of complete non-entity information in WkNER, the algorithm achieves a certain performance improvement on both the Chinese and English datasets used. Significant enhancements can be observed. This proves that the introduced word segmentation information can enhance the model's ability to predict entity boundaries.

Model	Weibo NER	OntoNotes 4.0	CoNLL 2003
BERT-Base	68.48	80.25	91.06
+kNN	69.67	80.91	91.54
+WkNER (CN)	**70.33**	**81.80**	**91.93**
RoBERTa-Base	69.01	80.44	–
+kNN	69.58	80.65	–
+WkNER (CN)	**69.95**	**82.47**	–
RoBERTa-Large	69.95	81.62	91.47
+kNN	70.78	81.68	91.52
+WkNER (CN)	**71.53**	**82.64**	**92.44**

Table 4: Ablation studies of word segmentation information. F1-scores are reported. kNN means kNN-NER (Wang et al., 2022) with hyperparameter setting of k=256. CN means the use of complete non-entity construction retrieval tables by WkNER.

4.4.2. Impact of Non-entity Information at Different Scales

We further investigated the impact of using different proportions of non-entity information to construct retrieval tables on the enhancement effect of the model. From Figure 4, we can observe that at 80% and below, the kNN-NER algorithm often achieves equal or better model enhancement effects, excessive introduction of non-entity information will

instead reduce the algorithm's enhancement effect. In addition, from Table 5, it can be observed that the number of non-entity category tokens in each dataset is far greater than the number of entity category tokens. This indicates that there is a large amount of redundant information in the non-entity category information, and the existence of this redundant information will interfere with the performance of the enhanced model. Therefore, appropriately pruning the scale of non-entity information when constructing retrieval tables, and increasing the proportion of entity information in the retrieval table, can not only better improve model performance but also reduce spatial complexity.

Dataset	Entity	Non-entity	Total
	Chinese		
OntoNotes 4.0	41203	481910	523113
MSRA	227630	1811065	2038695
Weibo NER	4951	71527	76478
	English		
CoNLL 2003	58415	232385	290800
OntoNotes 5.0	329046	2426430	2755476

Table 5: The distribution of entity and non-entity information in the complete Chinese and English datasets.

5. Conclusion

In this study, we propose WkNER which combines word segmentation information to improve the performance of the fine-tuned model. We experiment with the original kNN-NER framework and our proposed algorithm on several publicly available Chinese and English datasets, and our algorithm can achieve comparable or even better performance than previous state-of-the-art models. Moreover, in low-resource scenarios, our algorithm trained on only 20% of the dataset outperforms models trained on the complete dataset. We also explore the influence of using retrieval tables constructed from different scales of non-entity information on enhancing model performance. Experiments show that compared to using complete non-entity information, WkNER improves the performance of fine-tuned models more when using appropriate scale non-entity information of 80% or less. It can be seen that the addition of word segmentation information can significantly enhance the effectiveness of the baseline model.

6. Limitations

We discuss here the limitations of proposed WkNER. Firstly, as mentioned in the paper, although WkNER performs well on flat NER tasks, it cannot recognize nested and discontinuous entities. This is because each chunk obtained from our word segmentation operation is a continuous token sequence. Secondly, in the construction of the retrieval table phase, how to selectively trim non-entity information is also an important research challenge. The proportion of non-entity information in the retrieval table can to some extent affect the gain effect of WkNER. Future research can focus on optimizing the construction of retrieval tables to minimize resource consumption. Also, one can consider using a multi-granularity word segmentation method to improve our method and explore solutions for nested NER tasks.

7. Acknowledgments

This work is supported in part by the National Key Research and Development Program of China 2021YFB3101201, Natural Science Foundation of China under Grant 62372395, 62372396, and 62172349, National Natural Science Foundation of Hunan Province under Grant No. 2023JJ30597, the Research Foundation of Education Bureau of Hunan Province under Grant No. 21B0139, Open Project of the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences under Grant No. SYSKF2101.

8. Bibliographical References

Oren Boiman, Eli Shechtman, and Michal Irani. 2008. In Defense of Nearest-Neighbor based Image Classification. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society.

Chun Chen and Fang Kong. 2021. Enhancing Entity Boundary Detection for Better Chinese Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-Shot Named Entity Recognition via Contrastive Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2006–2013. IOS Press.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search Engine Guided Neural Machine Translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5133–5140. AAAI Press.

James Alistair Hammerton. 2003. Named Entity Recognition with Long Short-Term Memory. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 172–175. ACL.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer.

Han He and Jinho D. Choi. 2021. The Stem Cell Hypothesis: Dilemma behind Multi-Task Learning with Transformer Encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5555–5577. Association for Computational Linguistics.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient Nearest Neighbor Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5703–5714. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR*, abs/1508.01991.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data*, 7(3):535–547.

Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN Search Component to Pretrained Language Models for Better QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3424–3430. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kevin Knight, Ani Nenkova, and Owen Rambow, editors. 2016. *Neural Architectures for Named Entity Recognition*. The Association for Computational Linguistics.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020a. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6836–6842. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A Unified MRC Framework for Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5849–5859. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified Structure Generation for Universal Information Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5755–5772. Association for Computational Linguistics.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the Usage of Lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5951–5960. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 2121–2130. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021a. Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2782–2794. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. 2021b. A Trigger-Sense Memory Flow Framework for Joint Entity and Relation Extraction. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1704–1715. ACM / IW3C2.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt Locating and Typing for Named Entity Recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12492–12507. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019a. Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5326–5331. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019b. Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5326–5331. Association for Computational Linguistics.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages

2065–2075. Association for Computational Linguistics.

Buzhou Tang, Jianglu Hu, Xiaolong Wang, and Qingcai Chen. 2018. Recognizing Continuous and Discontinuous Adverse Drug Reaction Mentions from Social Media Using LSTM-CRF. *Wirel. Commun. Mob. Comput.*, 2018.

Shuhe Wang, Xiaoya Li, Yuxian Meng, Tianwei Zhang, Rongbin Ouyang, Jiwei Li, and Guoyin Wang. 2022. kNN-NER: Named Entity Recognition with Nearest Neighbor Search. *CoRR*, abs/2203.17103.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training Named Entity Tagger from Imperfect Annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5153–5162. Association for Computational Linguistics.

Shuang Wu, Xiaoning Song, and Zhen-Hua Feng. 2021. MECT: Multi-Metadata Embedding based Cross-Transformer for Chinese Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1529–1539. Association for Computational Linguistics.

Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting Multiple Embeddings for Chinese Named Entity Recognition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2269–2272. ACM.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A Unified Generative Framework for Various NER Subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6470–6476. Association for Computational Linguistics.

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, and Weiming Lu. 2022. De-Bias for Generative Extraction in Unified NER Task. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 808–818. Association for Computational Linguistics.

Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A Boundary-aware Neural Model for Nested Named Entity Recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 357–366. Association for Computational Linguistics.

Enwei Zhu and Jinpeng Li. 2022. Boundary Smoothing for Named Entity Recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7096–7108. Association for Computational Linguistics.

9. Language Resource References

Gina-Anne Levow. 2006. *The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition*. Association for Computational Linguistics. PID https://github.com/OYE93/Chinese-NLP-Corpus/tree/master/NER/MSRA.

Nanyun Peng and Mark Dredze. 2015. *Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings*. The Association for Computational Linguistics. PID https://github.com/OYE93/Chinese-NLP-Corpus/tree/master/NER/Weibo.

Sameer Pradhan and Alessandro Moschitti and Nianwen Xue and Hwee Tou Ng and Anders

Björkelund and Olga Uryupina and Yuchen Zhang and Zhi Zhong. 2013. *Towards Robust Linguistic Analysis using OntoNotes*. ACL, ISLRN 151-738-649-048-2.

Sameer Pradhan and Lance A. Ramshaw and Mitchell P. Marcus and Martha Palmer and Ralph M. Weischedel and Nianwen Xue. 2011. *CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes*. ACL, ISLRN 272-858-321-100-4.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. ACL, ISLRN 228-559-981-287-1.

A. Appendix

A.1. Segmentation Algorithm

Hanlp (He and Choi, 2021) can be directly used for Chinese data. For English data, it is necessary to first utilize the part-of-speech (POS) tagging function of Hanlp, and then customize a POS combination for each token to merge those tokens that can form entities into a single chunk. For example, neighboring words of a noun lexeme usually belong to the same entity, etc. The specific algorithm implementation is as follows:

Algorithm 2 Segmetation Algorithm

Input: Input sentence $\boldsymbol{x}=\{x_1, x_2, ..., x_{N-1}, x_N\}$;
Output: Output the segmented index w_{idx};

$pos = Hanlp.POS(\boldsymbol{x})$;
$i = 0$;
$w_{idx} = \emptyset$;
$e = \{$"$PROPN$", "$NOUN$", "NUM","ADJ"$\}$
while $i < len(\boldsymbol{x})$ **do**
  if $pos[i] \notin e$ **then**
    $i = i + 1$;
    $continue$;
  end if
  $j = i$;
  while $True$ **do**
    $j = j + 1$;
    if $j >= len(pos)$ **or** $(pos[j] \notin e - \{$"ADJ"$\}$ **and** $(pos[j] \neq$ "$PUNCT$" **or** $\boldsymbol{x}[j] \neq$ '$-$'$))$ **then**
      $w_{idx} = w_{idx} \cup \{[i, j-1]\}$;
      $break$;
    end if
  end while
  $i = j$;
end while
return w_{idx};

A.2. Datasets

Chinese OntoNotes 4.0 OntoNotes 4.0 (Pradhan et al., 2011) is a Chinese dataset, covering a variety of domain-specific language resources, including entity types: person, organization, location, etc.

Chinese MSRA MSRA (Levow, 2006) is a Chinese dataset collected from news domain texts. It contains three types of named entities and is used as shared task on SIGNAN backoff 2006.

Chinese Weibo NER Weibo NER (Peng and Dredze, 2015) is a Chinese dataset drawn from the social media website Sina Weibo and includes four types of named entities.

English CoNLL 2003 CoNLL 2003 (Sang and Meulder, 2003) is a English dataset used for named entity recognition tasks, with four named entity categories: person, organization, location, and others.

English OntoNotes 5.0 OntoNotes 5.0 (Pradhan et al., 2013) is a widely used English dataset in various fields, containing text data from different domains such as news reports, blog articles, social media posts, etc., which can be used to evaluate named entity recognition task models. It contains 18 types of named entities.

A.3. Hyperparameters

There are several hyperparameters involved in the experiment, that is, k, λ, τ, lr and $warmup\_proportion$. Among them, k represents the top k labels with the highest similarity to the token vectors in the kNN-NER and WkNER. And the method we utilize to select k is same as the experiment in Wang et al. (2022), so we based on their experimental results to directly choose parameters from the set {128, 256, 512} for experimentation. λ is hunted in [0, 1], indicating the influence degree of the model distribution with respect to the partition distribution and the kNN retrieval distribution in the final probability distribution. τ is a hyperparameter used to control the smoothness of the probability distribution, and we search for it between (0, 1]. The learning rate lr is usually chosen in {1e-5, 2e-5, 3e-5, 4e-5, 5e-5}, and $warmup\_proportion$ is chosen in the parameter set {1e-1, 1e-2, 1e-3}. The random seed in the experiment is selected from {42, 1204, 1660}.