Improving Low-Resource Sequence Labeling with Knowledge Fusion and Contextual Label Explanations

Anonymous ACL submission

Abstract

Sequence labeling remains a significant challenge in low-resource, domain-specific scenarios, particularly for character-dense languages. Existing methods primarily focus on enhancing model comprehension and improving data diversity to boost performance. However, these approaches still struggle with inadequate model applicability and semantic distribution biases in domain-specific contexts. To overcome these limitations, we propose a novel framework that combines an LLM-based knowledge enhancement workflow with a span-based Knowledge Fusion for Rich and Efficient Extraction (KnowFREE) model. Our workflow employs explanation prompts to generate precise contextual interpretations of target entities, effectively mitigating semantic biases and enriching the model's contextual understanding. The KnowFREE model further integrates extension label features, enabling efficient nested entity extraction without relying on external knowledge during inference. Experiments on multiple domain-specific sequence labeling datasets demonstrate that our approach achieves state-of-the-art performance, effectively addressing the challenges posed by low-resource settings.

1 Introduction

Sequence labeling is a fine-grained information extraction (IE) task that includes sub-tasks such as named entity recognition (NER), word segmentation, and part-of-speech (POS) tagging, playing a critical role in various downstream natural language processing (NLP) applications.

In low-resource scenarios, sequence labeling remains a persistent challenge, primarily due to the scarcity of domain-specific data, which limits the model's capacity to learn accurate label distributions. Moreover, character-dense languages such as Chinese pose additional difficulties, as the absence of explicit word boundaries greatly complicates label inference.

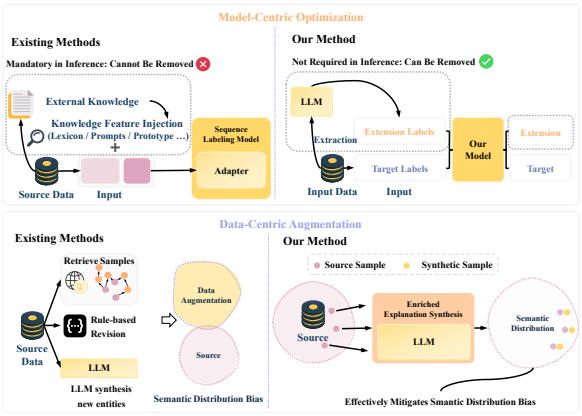

Figure 1: Distinctions between our method and existing methods in terms of model and data.

Previous studies predominantly focus on two main directions to enhance sequence labeling in low-resource scenarios: **(1) Model-Centric Optimization.** These methods focus on enhancing model's comprehension to detect implicit word boundaries and contextual signals through feature engineering. For instance, lexical features are injected via lexicon matching networks (Zhang and Yang, 2018a; Li et al., 2020; Liu et al., 2021; Wu et al., 2021) or prompt templates (Ma et al., 2022b; Shen et al., 2023; Chen et al., 2021b; Das et al., 2023) to strengthen entity boundary or type detection. Other methods employ knowledge transfer techniques such as Gaussian embeddings (Si et al., 2024; Das et al., 2022), prompt-based metrics (Chen et al., 2023; Lai et al., 2022), and contrastive learning (Huang et al., 2022; Zhang et al., 2024) to distill knowledge into target domains. **(2) Data-Centric Augmentation.** Meanwhile, data-centric methods concentrate on using data augmentation through altering entity label information (Hu et al., 2023; Yang et al., 2018), back translation (Paolini et al., 2021; Yaseen and Langer, 2021), and extracting knowledge from the external environment (Cai et al., 2023; Chen et al., 2021a; Yaseen and Langer,

2021) to enrich the dataset. With the advent of large language models (LLMs), recent findings leverage their generative capabilities to enhance the diversity of entity and sentence synthesis (Kang et al., 2024; Ye et al., 2024).

However, as illustrated in Figure 1, significant limitations remain when applying these solutions to specialized domains: **(1) Limited Model Applicability.** Existing model-centric approaches for character-dense languages often struggle to effectively incorporate diverse feature types and label structures, limiting the flexibility and expressiveness of feature injection. These methods also face difficulties in handling nested entities, further reducing their adaptability. Moreover, many approaches rely on rigid feature integration pipelines and complex input configurations to improve word features, leading to increased reliance on supplementary structures during inference and raising deployment costs. **(2) Variability in Label Distribution.** Existing data-centric augmentation methods frequently suffer from domain distribution biases. Inconsistencies in entity type definitions and semantic contexts across domains lead to mismatches in label priors and entity representations, undermining the quality of synthesized data and weakening zero-shot generalization.

These challenges, including structural rigidity and distributional mismatch, collectively hinder the practical effectiveness of current methods. This motivates our development of a unified framework that addresses both architectural constraints and domain adaptation challenges in a holistic manner. In this task, we adopt two key strategies for improving low-resource sequence labeling in character-dense languages: *(i) enhancing the utilization of non-entity features through the span-based model* and *(ii) improving the model's contextual understanding of target entities.*

To achieve these objectives, we propose a novel LLM-based data augmentation framework. Our approach begins by designing extraction prompts to identify and extract informative non-target entity features from the input text, thereby maximizing the utilization of non-entity information. ***To address the issue of limited model applicability,*** we introduce a span-based model called **Know**ledge **F**usion for **R**ich and **E**fficient **E**xtraction (**KnowFREE**), which supports nested entity annotation and integrates extension label features through a local multi-head attention module. Unlike previous methods, KnowFREE captures rich contextual representations during training without relying on external knowledge at inference time. ***To tackle the issue of variability in label distribution,*** we incorporate explanation prompts inspired by label explanation techniques (Golde et al., 2024; Yang and Katiyar, 2020; Ma et al., 2022a), enabling the generation of precise, context-aware explanations for target entities. This enhances the model's contextual understanding and mitigates semantic distribution biases. By leveraging LLMs for label interpretation synthesis, our framework outperforms other related data augmentation techniques in low-resource settings. We evaluate it on multiple Chinese and English domain-specific sequence labeling datasets, and experimental results demonstrate its effectiveness in overcoming the key limitations of low-resource scenarios.

The contributions of our work can be summarized as follows:

(1) *New method.* We propose a span-based KnowFREE model that supports nested label annotations and integrates multi-label features by a local multi-head attention module, which can be used without relying on external knowledge at inference.

(2) *New perspective.* To the best of our knowledge, we are the first to propose an approach that supports the seamless integration of extension label features within the model while eliminating the need for external features during inference.

(3) *State-of-the-art performance.* Experimental results demonstrate that our approach achieves outstanding performance on low-resource sequence labeling tasks.

2 Related Work

Span-based Sequence Labeling Methods Span-based sequence labeling methods have gained prominence for their ability to address overlapping and nested entities effectively. Early works, such as Dozat and Manning (2017); Yu et al. (2020) introduced Biaffine models to capture sentence-wide structures and score span boundaries for accurate information extraction. Based on this, Su et al. (2022) proposed the Global Pointer model, optimizing the Biaffine transformation's weight matrix and bias terms to boost efficiency and precision in span-based NER. In parallel, Shen et al. (2022) introduced a parallel instance query network for simultaneous entity extraction. Then, (Yan et al., 2023) proposed a multi-head Biaffine mechanism combined with CNNs to capture local span fea-

tures, achieving improved performance, while (Li et al., 2022) combined bilinear classifiers with dilated convolutions post-CLN to refine span-level relation classification. For few-shot NER, (Wang et al., 2022) introduced SpanProto, which integrates prototype-based classification with a contrastive loss to effectively separate non-target spans from prototype clusters, demonstrating strong performance in low-resource scenarios.

Sequence Labeling via LLMs

Recent advances in LLMs (OpenAI, 2023; DeepSeek-AI et al., 2025; Touvron et al., 2023) have introduced new paradigms for sequence labeling. Generative methods based on in-context learning (ICL) allow LLMs to perform labeling tasks directly without task-specific fine-tuning (Jiang et al., 2024). In zero-shot settings, InstructUIE (Wang et al., 2023) adopts a single-turn instruction framework across diverse IE tasks, UniversalNER (Mayhew et al., 2024) demonstrates improved performance by querying one entity type at a time, and GoLLIE (Sainz et al., 2024) enhances generalization via structured code-style prompting. LLMs have also shown promise in handling cross-domain and nested entity recognition (Nandi and Agrawal, 2024; Kim et al., 2024). In parallel, LLM-based data augmentation strategies synthesize high-quality training data by injecting domain-specific features (Ye et al., 2024; Heng et al., 2024), while others combine lightweight span detectors with LLM validation to improve span selection in specialized domains (Chen et al., 2024).

3 Method

In this section, we will introduce the knowledge enhancement workflow in § 3.1 and the specific structure of our sequence labeling KnowFREE model in § 3.2. The overall framework is shown in Figure 2.

3.1 Workflow of Knowledge Enhancement

In the knowledge enhancement workflow, we leverage LLMs to annotate potential entity information in the source sample and provide additional descriptions of entities. This enhances the utilization of non-entity features and improves the model's comprehension of the context in which the target entity appears. Our workflow consists of two main pipelines, which we describe in detail below.

Label Extension Annotation. In low-resource scenarios, non-entity segments in the sentence may contain additional non-target entity features, while data samples with flat-only entities may potentially contain nested entity information around the target entities. Leveraging these potential feature information can enhance the model's capacity to comprehend the fine-grained semantics and the ability to distinguish entity boundaries in character-dense languages. To achieve this, we utilize the LLM's general knowledge to generate extension entity tags, word segmentation tags, and part-of-speech (POS) tags for the source samples.

Formally, we can denote the source samples as $S = \{s_1, s_2, \ldots, s_n\}$, and the LLM as L, where n is the number of samples. We then constructed a prompt for each type of tag extraction, denoted as $\mathcal{P}_{ent}$, $\mathcal{P}_{seg}$, and $\mathcal{P}_{pos}$, correspondingly. The extension tags set can be computed as:

$$\hat{E}^{(k)} = \bigcup_{i=1}^{n} \hat{E}_i^{(k)}, \quad \hat{E}_i^{(k)} = L(s_i, \mathcal{P}_k), \quad (1)$$

where $\hat{E}_i^{(k)}$ represents the extension set using prompt $\mathcal{P}_k$ extracted from sentence s_i.

We assume that accurate and diverse extension tags can significantly improve the model's comprehension of context and its capacity for entity detection. Then, there are several issues that need to be solved in the results of extraction. (1) The uncertainty generation by LLMs leads to the presence of tags in the extension entity set in various textual formats yet denotes the same label category. Directly introducing these labels will significantly disrupt the model's assessment of the entity type. (2) Word segmentation usually produces boundary labels with low information entropy. However, POS tagging based on LLMs sometimes has missing annotations. Notably, POS tagging labels inherently include implicit word boundary information. Therefore, combining the outputs of these two processes is expected to yield better results.

To address the first issue, we use LLM to generate synonymous label mapping, and combine entity clustering algorithm to achieve synonymous label merging. Let denote the extension entity set as $\hat{E}^{\text{ent}} = \{(e_i, t_i) \mid e_i \in \mathcal{E}, t_i \in \mathcal{T}\}$, where $\mathcal{E}$ is the set of entities and $\mathcal{T}$ is the set of entity types. We compute the synonymous tag set by using the synonymous tag merge prompt $\mathcal{P}_{merge}$:

$$\mathcal{T}_s = L(\mathcal{T}, \mathcal{P}_{\text{merge}}), \quad (2)$$

$$\mathcal{T}_s = \{\tilde{T}_i : \{T_{i1}, T_{i2}, \ldots, T_{ir}\} \mid i \in [1, m]\}, \quad (3)$$

where $\tilde{T}_i$ is the standard label, T_{ij} is the synonymous label, and m, r is the number of standard

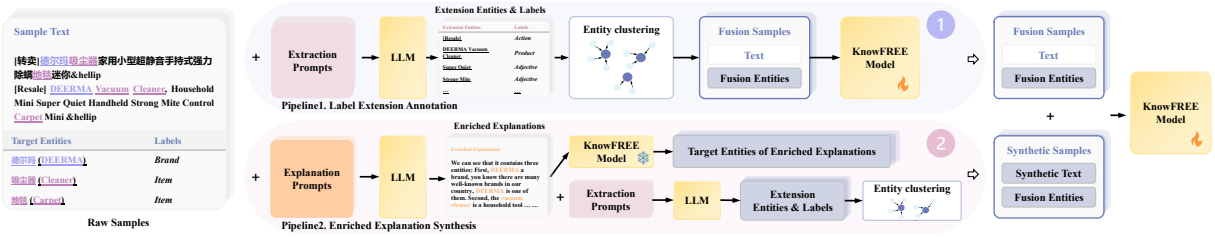

Figure 2: The workflow of our knowledge enhancement framework. Pipeline 1 generates extension entities to improve the KnowFREE performance. Pipeline 2 synthesizes additional training samples and entities, leveraging a frozen KnowFREE to annotate target entities.

label and its corresponding synonymous labels, respectively. This method is determined by the LLM's literal interpretation of the label, which may not accurately align the semantic spatial distribution of entities in the target domain, and therefore is not completely reliable. We further compute the vector representation of each entity-label pair by using a sentence embedding model $\mathcal{M}$, and the vector set of T_k can be represented as:

$$\mathbf{V}_k = \{\mathbf{v}_i \mid (e_i, t_i) \in \hat{E}^{\text{ent}}, \mathbf{t}_i \in T_k\}, \quad (4)$$

where $\mathbf{v}_i$ refers to $\mathcal{M}(x_i)$, x_i is the concatenation of e_i and t_i with the template of "$[e_i]$ is $[t_i]$". The center point and mean radius of the vector set for T_k can be calculated as:

$$\mathbf{c}_k = \frac{1}{|\mathbf{V}_k|} \sum_{\mathbf{v}_i \in \mathbf{V}_k} \mathbf{v}_i, \quad (5)$$

$$r_k = \frac{1}{p} \sum_{j=1}^{p} \|\mathbf{v}_j - \mathbf{c}_k\|, \quad (6)$$

where p denotes the Top-p samples that exhibit the greatest distance from $\mathbf{c}_k$. For each standard label $\tilde{T}_i$, we identify the synonymous label vector set $\mathbf{V}_{i,max}$ that contains the largest number of samples and has the largest radius, designating it as the reference vector set. We then evaluate whether each remaining synonymous label vector set $\mathbf{V}_{i,j}$ satisfies the condition $\|\mathbf{c}_j - \mathbf{c}_{max}\| \leq \epsilon \cdot r_{max}$. If the condition is met, T_j is merged into $\tilde{T}_i$; otherwise, T_j is treated as an independent standard label.

To address the second issue, we first compute the word segmentation set $\hat{E}^{\text{seg}}$ using $\mathcal{P}_{seg}$. Then, we combine $\hat{E}^{\text{seg}}$, $\mathcal{P}_{pos}$, and the original source sample as input to the LLM to generate part-of-speech tags without omissions. This approach ensures comprehensive POS tagging for all words in the sample while enhancing the diversity of word segmentation features.

Finally, we merge all the extension entities with standardized labels into the original data to obtain the fusion samples. We use our KnowFREE model to first train an annotation model on the fusion samples, which can be used for entity annotation in the next pipeline.

Enriched Explanation Synthesis. Injecting extension entity features has been proven to enhance model performance in many cases, its impact remains constrained in the following situations: (1) The datasets with short sentence contains a high proportion of target entities, leading to a relatively low amount of non-entity information in the sample. This results in a diminished validity of the external features introduced. (2) As the quantity of samples diminishes, their scarcity will emerge as the principal factor limiting performance enhancement. Enhancing the exploitation of non-entity features in the sample yields marginal performance enhancement. Both instances highlight the necessity of expanding the sample size. Nonetheless, the divergence in semantic distribution indicates that incorporating samples from outside the source domain, along with directly employing LLM to generate new sentences from existing entities, may introduce potential noise that could significantly affect model performance.

To tackle the aforementioned challenges, we employ LLM to generate entity explanations within the current samples, which aims to leverage the knowledge embedded within the LLM and the connections between samples and entities to produce the precise meaning of target entities within their context. This approach can mitigate noise caused by semantic distribution shifts in synthetic samples. Specifically, we define two types of explanation prompts: the Entity Explanation Prompt $\mathcal{P}_{exp}$ and the Extension Description Prompt $\mathcal{P}_{ext}$ for samples with and without target entities, respectively. The function of $\mathcal{P}_{exp}$ takes the source sample and

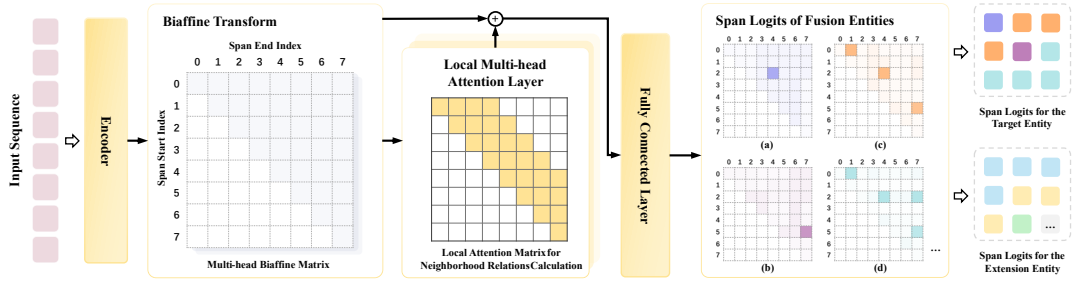

Figure 3: The architecture of the KnowFREE model. The span logits corresponding to the extension entity labels are ignored during inference.

its corresponding target entity as input, aiming to generate a detailed explanation of the entity's contextual meaning. Meanwhile, $\mathcal{P}_{ext}$ focuses on extracting and explaining key phrases from the text, with only the source sample as input. Additionally, to enrich the semantic representation of the source data, both prompts instruct the LLM to act the role of a domain expert, providing accessible and detailed explanations of the target entities to a hypothetical audience. This strategy encourages the LLM to generate comprehensive, contextually relevant, and easy-to-understand explanations, enhancing the overall semantic clarity of the dataset.

Next, we generate enriched explanations using the explanation prompts and proceed with annotating the entities in these synthetic texts through a two-branch pipeline: one for target entity annotation and the other for extension entity extraction. The frozen KnowFREE model, trained in the previous pipeline, is used to annotate the target entities. For extracting extension entities, we apply the same method as in the previous pipeline. Finally, all entities are integrated into the enriched explanations to produce synthetic samples. As shown in Figure 2, we can combine fusion samples and synthetic samples and retrain on the KnowFREE model to enhance its performance in the low-resource scenario.

3.2 Structure of KnowFREE Model

To support the fusion of multi-label knowledge, the KnowFREE model is built upon a Biaffine architecture, as illustrated in Figure 3. Unlike previous methods that rely on external feature injection, our approach eliminates the need for additional injection modules at the input stage.

Formally, let denote $X, \hat{X}$ as inputs of the fusion sample and the synthetic sample, respectively. The target entity spans of each sample is represented as $[s_i, e_i, l_i]$, where s_i, e_i are the start and end indices of the entity span, and l_i is the label type. The extension entity span is represented as $[s_j, e_j, \tilde{l}_j]$, where $\tilde{l}_j$ is the extension label type. We adopt a pre-trained encoder to compute the hidden states $H \in \mathbb{R}^{L\times D}$ for each input, where L is the length of the input sequence, D is the dimension of hidden states. We then compute the encoding of the entity's start and end positions:

$$H_s = \sigma(H\mathbf{W}_s), \quad H_e = \sigma(H\mathbf{W}_e), \tag{7}$$

where $\mathbf{W}_s, \mathbf{W}_e \in \mathbb{R}^{D\times D'}$ are learnable weight matrics, D' is the feature hidden size, and σ is the activation function. The Biaffine matrix of spans is computed by a multi-head Biaffine decoder $\mathcal{F}_{\text{MHB}}$ (Yu et al., 2020):

$$H_{\text{B}} = \mathcal{F}_{\text{MHB}}(H_s, H_e), \tag{8}$$

where $H_{\text{B}} \in \mathbb{R}^{L\times L\times \tilde{D}}$, $\tilde{D}$ is the hidden size of the Biaffine matrix. To improve the interactivity between multi-label features and span neighborhoods, we introduce a local multi-head attention layer to generate a mask for local multi-head attention. Each token is restricted to attend only within a local window of size ω through a masking scheme:

$$M[i,j] = \begin{cases} 0 & \text{if } |i-j| \le w, \\ -\infty & \text{otherwise.} \end{cases} \tag{9}$$

where $M \in \mathbb{R}^{L\times L}$. For input features H_{B}, the attention computation follows the standard multi-head paradigm with K heads, but incorporates the local mask M:

$$\mathcal{A}(Q,K,V,M) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V, \tag{10}$$

The outputs H_{attn} from all heads are concatenated and processed with layer normalization. We maintain training stability through residual connections:

$$H_{\text{G}} = \text{LayerNorm}(H_{\text{B}} + H_{\text{attn}}). \tag{11}$$

We then use the fully connected layer to map the sum of H_{G} and H_{B} into the number of entity tags:

$$P = \sigma^*(\mathbf{W}_O(H_{\mathrm{B}} + H_{\mathrm{G}}) + \mathbf{b}), \quad (12)$$

where $\mathbf{W}_O \in \mathbb{R}^{\tilde{D}\times(N_{\text{tgt}}+N_{\text{ext}})}$ is the learnable weight matrix, σ^* is the activation function, and $\mathbf{b} \in \mathbb{R}^{(N_{\text{tgt}}+N_{\text{ext}})}$ is the bias. $N_{\text{tgt}}, N_{\text{ext}}$ are the number of target entity label and extension entity label, respectively. The binary cross entropy is employed to compute the loss. To prevent the model from overly concentrating on features of extension entities, the loss function is defined as:

$$\mathcal{L} = -(\sum_{\substack{0\le i,\\ j<N_{\text{tgt}}}} y_{i,j}\log P_{i,j} + \alpha \sum_{\substack{N_{\text{tgt}}\le i,\\ j<N_{\text{ext}}}} y_{i,j}\log P_{i,j}), \quad (13)$$

where y refers to the ground truth labels, α is the weight parameter. Similarly, we control the influence of the quantity and noise in synthetic samples with another weight parameter. The final loss is a weighted sum of the original and synthetic losses:

$$\mathcal{L}_{\text{final}} = \mathcal{L} + \beta\mathcal{L}_{syn}. \quad (14)$$

During inference, the weights associated with the extension entity labels are masked, ensuring that the model exclusively predicts the target entities. This design simplifies the overall architecture while enhancing model efficiency.

4 Experiment

4.1 Experiment Setup

Training: To comprehensively evaluate the effectiveness of our data augmentation strategy on different LLMs, and to fairly compare previous related methods. We use ChatGLM3-6B, GLM4-9B-Chat, Qwen2.5-14B-Instruct, Llama3.1-70B-Instruct, GPT-4o and Deepseek-V3 (GLM et al., 2024; Yang et al., 2024b,a; Dubey et al., 2024; OpenAI, 2023; DeepSeek-AI et al., 2025) as LLMs for label extension annotation and enriched explanation synthesis. We choose BERT (Devlin et al., 2019) as the backbone encoder of the KnowFREE. More detail settings are presented in Appendix H.

Evaluation: To assess the performance of our method in low-resource scenarios, we conducted experiments on a variety of datasets. These include Chinese flat NER datasets (Weibo (Peng and Dredze, 2015), Youku (Jie et al., 2019), Taobao (Jie et al., 2019), and Resume (Zhang and Yang, 2018b)); English flat NER datasets (CoNLL'03 (Sang and De Meulder, 2003) and MIT-Movie (Liu et al., 2013)); a Chinese nested NER dataset (CMeEE-v2 (Zhang et al., 2022)); word segmentation datasets (PKU and MSR (Emerson, 2005)); and a POS tagging dataset (UD (Nivre et al., 2016)). To evaluate our method under data scarcity and explore the limits of performance gains from sample synthesis, we conducted both **many-shot** and **few-shot** experiments. In the many-shot setting, we simulated low-resource conditions by randomly sampling subsets of 250, 500, and 1000 training instances. In the few-shot setting, we adopted the standard "n-way k-shot" paradigm, using a greedy sampling strategy to ensure each target label appeared at least k times. To ensure consistency, each larger subset included all samples from the smaller ones. Dataset statistics are provided in Appendix G. We then discuss the effectiveness of each module in the analysis section. Additionally, we report the results on the full datasets in Appendix B, conduct further ablation studies on the number of heads in local attention in Appendix D, and provide visualizations of the logit scores for the extension labels in Appendix F.

Baselines: To ensure a fair comparison, we evaluate our method against both model-centric and data-centric baselines. On the model-centric side, we compare with general baselines such as BERT-CRF (Devlin et al., 2019) and BiaffineNER (Yu et al., 2020), as well as models specifically designed for Chinese sequence labeling, including FLAT (Li et al., 2020), MECT (Wu et al., 2021), and LEBERT (Liu et al., 2021). We also include comparisons with state-of-the-art nested NER methods such as W^{2}NER (Li et al., 2022), CNN Nested NER (Yan et al., 2023), and DiFiNet (Cai et al., 2024). From the data-centric perspective, we compare with the transfer learning approach PCBERT (Lai et al., 2022), and LLM-enhanced methods including LLM-DA (Ye et al., 2024), ProgGen (Heng et al., 2024), and MELM (Zhou et al., 2022). In addition, Appendix A provides results comparing vanilla LLMs and LoRA fine-tuned models on sequence labeling tasks.

4.2 Main Results

Many-shot Results. As shown in Table 1, our method consistently achieves higher average performance across all datasets. We observe that larger backbone LLMs generally bring greater performance improvements to our approach. Although the scaling law is not strictly linear, even the smallest model, ChatGLM3-6B, delivers strong results.

Model	Weibo			Youku			Taobao			Resume			CMeEE-v2			PKU			MSR			UD		
	250	500	1000	250	500	1000	250	500	1000	250	500	1000	250	500	1000	250	500	1000	250	500	1000	250	500	1000
BERT-CRF (Devlin et al., 2019)	56.57	60.91	66.52	68.02	70.57	74.92	68.78	71.88	74.74	90.19	92.35	93.43	-	-	-	93.49	94.31	95.02	90.60	91.73	92.93	87.11	89.87	91.98
FLAT (Li et al., 2020)	57.75	59.47	65.72	72.31	76.01	78.73	69.84	71.72	76.21	91.35	93.04	93.61	-	-	-	78.28	80.10	80.22	77.72	78.27	78.44	77.76	78.39	80.11
MECT (Wu et al., 2021)	58.55	60.77	66.13	72.82	75.85	79.16	70.54	73.87	76.48	91.52	93.63	93.90	-	-	-	87.28	87.54	87.58	87.48	87.53	87.71	87.12	87.30	87.61
LEBERT (Liu et al., 2021)	61.23	64.03	67.63	72.39	75.00	77.88	71.12	74.46	77.44	93.08	94.16	95.05	-	-	-	93.42	94.09	94.97	90.64	91.95	93.27	88.93	91.85	93.43
PCBERT (Lai et al., 2022)	70.73	70.78	72.81	77.67	81.96	83.66	73.32	75.41	79.21	93.63	94.31	95.18	-	-	-	93.47	94.07	94.54	90.01	92.18	93.10	89.81	91.70	93.67
BiaffineNER (Yu et al., 2020)	58.74	66.41	69.70	77.28	80.21	81.68	74.62	76.98	79.46	93.30	94.61	95.49	61.26	65.56	68.40	93.20	94.39	94.94	91.60	92.63	93.64	88.84	90.74	92.68
W^{2}NER (Li et al., 2022)	54.57	63.21	71.09	79.20	81.40	83.28	74.68	76.80	79.71	94.39	95.82	96.35	61.10	65.67	68.72	93.90	94.64	95.41	91.61	92.76	93.82	90.12	92.91	94.86
CNN Nested NER (Yan et al., 2023)	64.81	67.75	69.96	79.28	81.58	83.94	75.39	77.86	80.06	93.10	94.54	95.39	62.32	66.43	69.06	94.00	94.59	95.47	91.72	92.86	93.67	90.21	92.48	94.70
DiFiNet (Cai et al., 2024)	67.35	69.02	72.19	79.81	81.32	83.29	75.40	77.05	79.61	93.81	94.75	95.93	63.55	66.26	67.28	93.81	94.76	95.26	91.46	92.45	93.24	90.94	92.82	94.62
KnowFREE-F (ChatGLM3-6B)	66.76	71.59	72.99	79.30	82.13	84.50	76.31	**78.55**	80.53	94.03	95.04	96.14	63.64	67.47	69.52	94.07	94.94	95.51	91.73	92.91	93.92	90.99	92.71	95.00
KnowFREE-F (GLM4-9B-Chat)	66.40	73.08	72.59	79.40	82.16	84.37	76.02	78.21	80.48	94.05	95.43	96.25	63.98	67.41	69.38	94.57	95.01	95.49	91.83	92.73	93.91	90.58	92.72	94.98
KnowFREE-F (Qwen-14B)	68.07	73.39	73.41	80.30	82.10	84.20	76.38	78.00	80.50	94.21	95.32	96.40	63.23	67.25	69.19	94.36	94.94	95.51	91.76	92.93	93.89	90.53	93.03	94.97
KnowFREE-F (Llama3.1-70B-Instruct)	67.72	73.08	72.59	80.18	82.17	84.37	76.24	78.32	80.51	94.11	95.24	96.18	63.92	67.47	69.18	94.28	94.98	95.51	91.77	92.89	93.91	90.76	92.69	94.96
KnowFREE-F (GPT-4o)	68.18	73.51	74.06	80.30	82.18	84.62	76.47	78.18	**80.64**	94.55	95.50	96.37	63.66	67.59	69.76	94.59	95.09	**95.62**	91.97	92.96	**93.98**	91.01	93.13	95.05
KnowFREE-F (Deepseek-V3)	68.12	73.42	74.12	80.28	82.13	84.39	76.29	78.19	80.50	94.50	95.49	96.42	63.79	67.51	69.73	94.52	95.04	95.61	91.79	92.97	93.88	91.10	93.30	95.12
KnowFREE-FS (ChatGLM3-6B)	**74.78**	77.18	76.78	80.29	83.09	84.45	76.49	77.94	79.54	94.71	95.40	96.18	66.80	68.67	69.29	94.54	95.09	95.50	92.11	93.01	93.67	92.09	93.65	94.77
KnowFREE-FS (GLM4-9B-Chat)	73.90	76.86	76.57	81.86	83.17	84.48	76.47	77.89	79.36	94.95	95.45	96.21	68.12	68.45	68.85	94.73	95.05	95.47	92.18	92.96	93.50	91.23	93.02	93.42
KnowFREE-FS (Qwen-14B)	73.09	73.96	74.09	81.39	82.82	83.71	76.48	77.83	79.19	94.55	95.33	96.06	66.58	67.61	68.53	94.59	95.04	95.48	92.47	93.14	93.59	92.73	93.88	94.36
KnowFREE-FS (Llama3.1-70B-Instruct)	74.18	76.91	76.68	81.69	83.09	84.36	76.48	77.91	79.69	94.86	95.46	96.21	68.12	68.45	68.85	94.62	95.02	95.47	92.16	93.02	93.66	92.68	93.76	94.23
KnowFREE-FS (GPT-4o)	74.25	77.12	77.16	**81.98**	83.16	84.48	76.47	78.12	80.02	94.59	95.49	96.31	68.31	**68.73**	**69.78**	94.72	**95.10**	95.51	92.62	93.16	93.96	**93.72**	**93.98**	95.33
KnowFREE-FS (Deepseek-V3)	74.77	**77.19**	**77.18**	81.72	**83.24**	**84.97**	**76.52**	78.21	80.56	**94.97**	**95.51**	**96.46**	**68.51**	68.40	69.12	**94.93**	95.02	95.49	**92.83**	**93.17**	93.42	93.18	93.81	**95.35**

Table 1: The overall results on many-shot sequence labeling tasks. KnowFREE-F denotes the variant using only the label extension annotation pipeline, while KnowFREE-FS incorporates the enriched explanation synthesis pipeline. The **bold** values indicate the best performance, and the underlined values represent the second-best.

Under the 250-sample setting, our method surpasses the strongest baseline by an average of 1.95%, especially with a 4.05% gain on the Weibo dataset. In low-resource settings, KnowFREE-FS outperforms KnowFREE-F. However, as the number of training samples increases, especially beyond 500, the performance of KnowFREE-FS becomes comparable to or slightly lower than that of KnowFREE-F. This indicates that enriched data synthesis is more effective when training data is limited. When more data is available, the noise introduced by synthetic samples may outweigh the benefits. Further analysis of noise effects is provided in Appendix F and E. Even when data synthesis becomes less effective in higher-resource scenarios, KnowFREE-F maintains strong performance. With 1000 training samples, it still outperforms the strongest baseline by an average of 0.95% across all NER datasets, demonstrating the robustness and effectiveness of the label extension annotation strategy.

Few-shot Results. To assess the effectiveness of our method in few-shot settings, we compared it with state-of-the-art nested NER models and several LLM-based data augmentation strategies, including LLM-DA, ProgGen, and MELM, on both Chinese and English NER datasets. For fair comparison, all synthesized samples were annotated using the KnowFREE model trained solely on the original data, and the resulting data were used to retrain KnowFREE. As shown in Table 2, our method consistently outperforms the baselines under few-shot settings. On the Weibo dataset with k=5, while other methods yield zero performance, our approach achieves the performance of 35.58%. Moreover, for $k \leq 15$, LLM-based augmentation strategies often perform worse than CNN Nested NER and DiFiNet, indicating limited domain adaptability and the adverse effects of noise introduced by data synthesis. The performance gains are more pronounced on Chinese datasets compared to English ones, demonstrating the method's robustness across languages and its particular strength

Size	**Ours**	LLM-DA	ProgGen	MELM	C.N.-NER	DiFiNet
				Weibo		
k=5	**35.58 (2.33)**	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
k=10	**48.53 (1.57)**	0.95 (0.23)	0.48 (0.19)	0.03 (0.04)	0.95 (0.21)	0.98 (0.18)
k=15	**60.61 (2.58)**	17.28 (1.93)	12.36 (1.22)	2.98 (0.58)	20.68 (1.88)	27.43 (2.30)
k=20	**68.62 (2.04)**	39.49 (2.41)	34.16 (2.09)	16.71 (1.67)	31.10 (2.01)	32.71 (2.31)
				Youku		
k=5	**38.76 (2.30)**	12.03 (2.24)	13.76 (1.32)	9.98 (2.01)	24.70 (2.55)	23.83 (2.52)
k=10	**68.95 (3.85)**	33.10 (2.34)	33.52 (2.01)	16.17 (1.30)	46.39 (2.48)	46.62 (2.43)
k=15	**71.76 (3.33)**	59.72 (3.94)	56.55 (1.56)	50.18 (1.67)	60.41 (1.88)	60.61 (1.91)
k=20	**72.83 (2.29)**	64.58 (2.53)	61.80 (1.60)	52.38 (1.59)	67.38 (1.92)	68.38 (1.58)
				Taobao		
k=5	**62.96 (2.34)**	13.95 (2.32)	22.74 (2.62)	8.97 (0.88)	22.40 (1.61)	23.82 (1.56)
k=10	**64.64 (3.58)**	54.05 (2.92)	50.75 (1.66)	32.41 (1.37)	53.33 (2.01)	53.75 (2.03)
k=15	**69.26 (2.65)**	59.69 (2.58)	59.77 (1.51)	55.32 (1.57)	60.45 (1.83)	61.01 (1.54)
k=20	**68.93 (2.36)**	61.79 (1.51)	61.77 (1.63)	43.13 (1.62)	63.36 (1.86)	64.08 (1.53)
				Resume		
k=5	**65.28 (0.97)**	30.44 (0.66)	27.63 (0.34)	20.67 (1.26)	25.50 (1.22)	29.97 (1.17)
k=10	**78.89 (0.53)**	45.40 (0.58)	50.39 (0.82)	42.34 (1.28)	49.78 (1.53)	50.19 (1.49)
k=15	**85.43 (1.17)**	58.77 (1.15)	60.32 (1.26)	53.18 (1.18)	56.04 (1.31)	56.92 (1.16)
k=20	**85.56 (1.11)**	75.13 (1.36)	82.96 (1.28)	69.15 (1.21)	67.51 (1.17)	68.83 (1.19)
				CMeEE-v2		
k=5	**49.68 (1.89)**	35.03 (1.65)	39.18 (1.52)	12.78 (1.01)	6.49 (0.56)	5.62 (0.47)
k=10	**60.46 (1.67)**	48.91 (1.61)	44.80 (1.59)	32.76 (1.65)	47.40 (1.56)	47.25 (1.50)
k=15	**62.46 (1.51)**	48.97 (1.54)	49.32 (1.61)	38.79 (1.53)	48.90 (1.65)	48.72 (1.59)
k=20	**63.83 (1.68)**	57.18 (1.63)	57.76 (1.51)	50.12 (1.67)	56.38 (1.58)	56.22 (1.55)
				CoNLL'03		
k=5	**64.18 (1.62)**	57.84 (1.93)	57.11 (2.56)	30.00 (2.13)	27.75 (1.61)	26.86 (1.36)
k=10	**75.83 (1.52)**	69.07 (2.44)	69.10 (2.39)	63.48 (2.18)	62.93 (1.94)	59.17 (1.88)
k=15	**78.68 (1.39)**	78.18 (2.22)	78.32 (2.19)	76.16 (2.03)	75.93 (1.86)	75.85 (1.89)
k=20	**83.24 (1.21)**	81.94 (2.17)	82.09 (1.55)	79.41 (2.21)	77.92 (1.58)	77.74 (1.81)
				MIT-Movie		
k=5	**57.34 (1.88)**	53.97 (2.07)	52.82 (2.44)	38.49 (2.23)	36.81 (1.26)	37.62 (1.33)
k=10	**64.08 (1.66)**	63.03 (2.35)	63.41 (2.14)	50.56 (2.29)	49.08 (1.60)	49.43 (1.58)
k=15	**67.03 (1.68)**	65.77 (1.46)	65.93 (1.17)	58.32 (1.87)	58.03 (1.69)	58.54 (1.51)
k=20	**69.28 (1.63)**	69.12 (1.08)	69.19 (1.59)	62.02 (1.91)	60.81 (1.60)	61.07 (1.64)

Table 2: Results of few-shot sequence labeling tasks. Our default method is KnowFREE-FS (ChatGLM3-6B). C.N.-NER refers to the abbreviation of CNN Nested NER. Values in parentheses indicate standard deviation. **bold** numbers highlight the best performance.

in character-dense languages. Further analysis of performance trends of LLM-based methods under varying data sizes is provided in Appendix E.

4.3 Analysis

Method	Weibo	Youku	Taobao	Resume	CMeEE-v2	PKU	MSR	UD
Default	**72.99**	**84.50**	**80.53**	**96.14**	**69.52**	**95.51**	93.92	95.00
w/o L.E.A.	72.32	84.26	79.93	96.02	69.49	93.75	93.67	94.88
w/o local attn & L.E.A.	69.87	81.65	79.01	95.49	68.42	94.93	93.64	92.66
w/o local attn w cnn	72.21	84.28	80.26	95.94	69.31	95.50	93.72	94.76
w/o S.L.	72.26	84.19	79.80	95.74	69.18	94.86	93.87	94.85
w/o entity	72.25	84.37	80.14	95.86	69.08	95.51	93.91	94.97
w/o pos	72.39	84.45	80.48	**96.14**	69.26	**95.51**	**93.93**	**95.01**

Table 3: Results of F1 scores in ablation studies, all results are trained on datasets with 1000. The backbone LLM is ChatGLM3-6B.

Ablation Studies: To evaluate the contribution of each component in our approach, we conducted ablation studies by selectively removing modules and analyzing their impact on model performance, as shown in Table 3. "w/o L.E.A." removes the Label Extension Annotation module and uses the vanilla KnowFree model. Although this leads to a performance drop, it still outperforms nested NER baselines across several datasets. "w/o local attn & L.E.A." disables both the local attention and L.E.A. modules, resulting in a significant average performance drop of 1.56%, highlighting their combined effectiveness. In "w/o local attn w CNN," the local attention module is replaced by the masked CNN module from CNN Nested NER. While this improves performance over CNN Nested NER, it underperforms compared to our attention-based model, confirming the advantage of local multi-head attention for capturing neighborhood interactions. "w/o S.L." removes the synonymous label merging strategy and causes a 0.42% drop in performance, indicating that failing to unify semantically equivalent labels introduces confusion and weakens model predictions. In "w/o entity" and "w/o pos," we exclude extension entity features and POS features, respectively. Removing entity features leads to a larger performance drop, showing their stronger impact on entity recognition. Interestingly, removing POS features improves results on MSR and UD, possibly due to noise introduced by imperfect or overly correlated POS tags.

Impact of Enriched Explanation Synthesis in K-Shot Sampling: To further evaluate the impact of enriched explanation synthesis on model performance in low-resource scenarios, we conducted experiments following the "n-way k-shot" paradigm. The sampled data was then augmented with ChatGLM3-6B. We compared the model's performance with and without enriched explanation synthesis, as shown in Figure 4. Here, "w Synthetic" indicates the performance with enriched explanation synthesis, while "w/o Synthetic" reflects the performance without it. The results demonstrate that "w Synthetic" achieves a substantial performance boost over "w/o Synthetic" from the outset. Notably, when k is less than 15, enriched explanation synthesis consistently delivers rapid performance improvements across all datasets. As k increases, the performance gap narrows, but "w Synthetic" continues to outperform "w/o Synthetic" across all settings. These findings highlight that in resource-scarce scenarios, synthesizing enriched data is more effective than directly injecting features into raw samples. These findings highlight the critical role of enriched explanation synthesis in enhancing model performance, particularly when labeled data is limited.

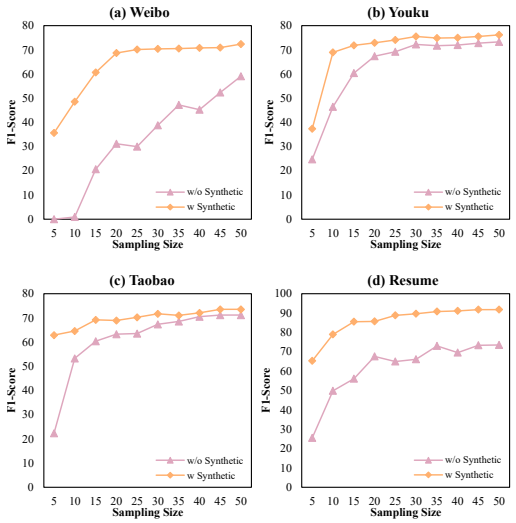

Figure 4: Performance comparison with and without enriched explanation synthesis under k-shot sampling.

5 Conclusion

In this paper, we propose a novel framework that integrates an LLM-based knowledge enhancement workflow with a span-based sequence labeling model. Our approach improves model performance by generating contextual interpretations of target entities and annotating extension labels. Additionally, our KnowFREE model effectively incorporates extension label features to enhance extraction capabilities. Extensive experiments on Chinese few-shot sequence labeling datasets demonstrate that our method achieves state-of-the-art performance, showcasing its effectiveness and efficiency.

Limitations

While enriched explanation synthesis significantly improves model performance in low-resource scenarios (e.g., with fewer than 500 original samples), its effectiveness diminishes as the size of the original dataset increases. Specifically, when the number of original samples exceeds this threshold, distributional discrepancies between synthetic samples and target domain semantics can lead to the synthetic data having a negative impact that outweighs its benefits. In future work, we plan to explore adaptive alignment mechanisms to better align synthetic and original data across different data scales.

Ethics Statement

Our data augmentation method utilizes LLMs to generate data independently of the existing training set. However, the generated data may reflect social biases inherent in the pre-training corpus. To mitigate the risk of propagating biased information into sequence labeling models, we recommend conducting manual reviews before integrating the synthesized data into practical applications.

References

Jiong Cai, Shen Huang, Yong Jiang, Zeqi Tan, Pengjun Xie, and Kewei Tu. 2023. Improving low-resource named entity recognition with graph propagated data augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 110–118, Toronto, Canada. Association for Computational Linguistics.

Yuxiang Cai, Qiao Liu, Yanglei Gan, Run Lin, Changlin Li, Xueyi Liu, Da Luo, and JiayeYang JiayeYang. 2024. Difinet: Boundary-aware semantic differentiation and filtration network for nested named entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6455–6471. Association for Computational Linguistics.

Shuguang Chen, Gustavo Aguilar, Leonardo Neves, and Thamar Solorio. 2021a. Data augmentation for cross-domain named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5346–5356, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Chen, Lili Zhao, Zhi Zheng, Tong Xu, Yang Wang, and Enhong Chen. 2024. Double-checker: Large language model as a checker for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 3172–3181. Association for Computational Linguistics.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2021b. Lightner: A lightweight tuning paradigm for low-resource NER via pluggable prompting. *CoRR*, abs/2109.00720.

Yanru Chen, Yanan Zheng, and Zhilin Yang. 2023. Prompt-based metric learning for few-shot NER. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7199–7212. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6338–6353. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Haoran Zhang, Peng Shi, Wenpeng Yin, and Rui Zhang. 2023. Unified low-resource sequence labeling by sample-aware dynamic sparse finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6998–7010. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen,

Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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*, pages 4171–4186. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2005, Jeju Island, Korea, 14-15 October, 2005*. ACL.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Jonas Golde, Felix Hamborg, and Alan Akbik. 2024. Large-scale label interpretation learning for few-shot named entity recognition. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2915–2930, St. Julian's, Malta. Association for Computational Linguistics.

Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang. 2024. Proggen: Generating named entity recognition datasets step-by-step with self-reflexive large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 15992–16030. Association for Computational Linguistics.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S. Yu. 2023. Entity-to-text based data augmentation for various named entity recognition tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9072–9087, Toronto, Canada. Association for Computational Linguistics.

Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. 2022. COPNER: contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of*

Korea, October 12-17, 2022, pages 2515–2527. International Committee on Computational Linguistics.

Guochao Jiang, Zepeng Ding, Yuchen Shi, and Deqing Yang. 2024. P-ICL: point in-context learning for named entity recognition with large language models. *CoRR*, abs/2405.04960.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 729–734, Minneapolis, Minnesota. Association for Computational Linguistics.

Hyeonseok Kang, Hyein Seo, Jeesu Jung, Sangkeun Jung, Du-Seong Chang, and Riwoo Chung. 2024. Guidance-based prompt data augmentation in specialized domains for named entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 665–672. Association for Computational Linguistics.

Hongjin Kim, Jai-Eun Kim, and Harksoo Kim. 2024. Exploring nested named entity recognition with large language models: Methods, challenges, and insights. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8653–8670. Association for Computational Linguistics.

Peichao Lai, Feiyang Ye, Lin Zhang, Zhiwei Chen, Yanggeng Fu, Yingjie Wu, and Yilei Wang. 2022. PCBERT: Parent and child bert for chinese few-shot ner. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2199–2209.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10965–10973. AAAI Press.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuan-Jing Huang. 2020. FLAT: Chinese ner using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced chinese sequence labeling using BERT adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5847–5858. Association for Computational Linguistics.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022a. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971, Dublin, Ireland. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022b. Template-free prompt tuning for few-shot NER. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Suppa, Hila Gonen, Joseph Marvin Imperial, Börje Karlsson, Peiqin Lin, Nikola Ljubesic, Lester James V. Miranda, Barbara Plank, Arij Riabi, and Yuval Pinter. 2024. Universal NER: A gold-standard multilingual named entity recognition benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4322–4337. Association for Computational Linguistics.

Xu Ming. 2022. text2vec: A tool for text to vector.

Subhadip Nandi and Neeraj Agrawal. 2024. Improving few-shot cross-domain named entity recognition by instruction tuning a word-embedding based retrieval augmented large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*, pages 686–696. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774:1–100.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai,

Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 548–554. The Association for Computational Linguistics.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

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.

Yongliang Shen, Xiaobin Wang, Zeqi Tan, Guangwei Xu, Pengjun Xie, Fei Huang, Weiming Lu, and Yueting Zhuang. 2022. Parallel instance query network 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 947–961. Association for Computational Linguistics.

Shuzheng Si, Helan Hu, Haozhe Zhao, Shuang Zeng, Kaikai An, Zefan Cai, and Baobao Chang. 2024. Improving the robustness of distantly-supervised named entity recognition via uncertainty-aware teacher learning and student-student collaborative learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5533–5546, Bangkok, Thailand. Association for Computational Linguistics.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global pointer: Novel efficient span-based approach for named entity recognition. *CoRR*, abs/2208.03054.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971:1–27.

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022. SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3476, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shuang Wu, Xiaoning Song, and Zhenhua 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 (Volume 1: Long Papers)*, pages 1529–1539, Online. Association for Computational Linguistics.

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2023. An embarrassingly easy but strong baseline for nested named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1442–1452. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu

Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Usama Yaseen and Stefan Langer. 2021. Data augmentation for low-resource named entity recognition using backtranslation. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 352–358, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. LLM-DA: data augmentation via large language models for few-shot named entity recognition. *CoRR*, abs/2402.14568.

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.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhifang Sui, Baobao Chang, Hui Zong, Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. 2022. CBLUE: A chinese biomedical language understanding evaluation benchmark. 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 7888–7915. Association for Computational Linguistics.

Shan Zhang, Bin Cao, and Jing Fan. 2024. KCL: Few-shot named entity recognition with knowledge graph and contrastive learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9681–9692, Torino, Italia. ELRA and ICCL.

Yue Zhang and Jie Yang. 2018a. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1554–1564. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018b. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1554–1564. Association for Computational Linguistics.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: data augmentation with masked entity language modeling for low-resource NER. 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 2251–2262. Association for Computational Linguistics.

A Sequence Labeling with LLMs

Model	Weibo	Youku	Taobao	Resume	CMeEE	UD	PKU	MSR
ChatGLM3-6B	3.28	9.84	9.21	13.71	2.41	1.15	3.76	6.69
Llama3-8B-Instruct	15.06	13.89	9.67	41.45	11.27	14.06	23.88	28.84
GLM4-9B-Chat	12.43	31.69	14.90	47.12	16.28	26.38	**26.18**	**31.12**
Qwen2.5-14B-Instruct	31.06	**51.00**	5.89	51.67	22.38	10.06	2.33	3.02
Llama3.1-70B-Instruct	**34.69**	49.54	13.77	**66.60**	40.86	37.82	3.96	7.44
Deepseek-V3	33.33	10.91	**15.91**	60.88	**42.20**	**38.66**	5.96	2.45

Table 4: Results on sequence labeling tasks with vanilla LLMs on zero-shot learning.

To evaluate the performance of directly using LLMs for sequence labeling tasks in low-resource scenarios, we present the results of vanilla LLMs in Table 4 and the performance of LoRA-finetuned models trained on sampled datasets in Table 5. During zero-shot inference, LLMs extract entities based on the input text and target labels. As shown in Table 4, vanilla LLMs exhibit significantly poor performance on all datasets, likely due to their limited understanding of target label definitions. Moreover, models with different parameter scales show varying performance across datasets, and no clear positive correlation is observed between model size and performance. This may be improved through more advanced designs of the sequence labeling prompts used in our setting. Therefore, for domain-specific sequence labeling tasks, incorporating few-shot examples into the prompt or fine-tuning the model with LoRA could be a more effective and practical approach.

Model	Weibo			Youku			Taobao			Resume		
	250	500	1000	250	500	1000	250	500	1000	250	500	1000
ChatGLM3-6B (LoRA)	12.29	20.44	28.22	34.52	38.62	39.29	29.91	30.02	30.03	39.92	40.76	39.98
GLM4-9B-Chat (LoRA)	49.20	54.54	**60.30**	**73.79**	74.56	74.84	58.84	**63.63**	**69.20**	83.31	86.24	88.27
Llama3-8B-Instruct (LoRA)	10.09	17.08	44.24	49.59	60.97	64.33	23.84	34.65	42.98	65.36	70.82	74.86
Qwen2.5-14B-Instruct (LoRA)	**50.93**	**56.87**	56.30	72.89	**75.10**	**78.83**	**61.58**	59.43	66.98	**84.45**	**88.12**	**88.65**
Model	**CMeEE-v2**			**PKU**			**MSR**			**UD**		
	250	500	1000	250	500	1000	250	500	1000	250	500	1000
ChatGLM3-6B (LoRA)	12.75	13.30	13.55	20.62	26.27	29.21	23.53	26.04	29.38	32.20	37.57	37.81
GLM4-9B-Chat (LoRA)	50.67	**54.50**	56.22	71.41	**73.68**	75.02	75.70	78.64	**80.94**	72.32	**77.50**	79.69
Llama3-8B-Instruct (LoRA)	23.27	41.52	44.41	62.50	64.53	68.88	66.17	68.52	69.72	32.16	35.51	44.85
Qwen2.5-14B-Instruct (LoRA)	**53.59**	53.80	**59.31**	**71.94**	72.71	**75.91**	**77.83**	**79.89**	80.07	**74.40**	77.04	**80.70**

Table 5: Performance of LLM fine-tuning with LoRA on sequence labeling tasks.

In the experiments with LoRA fine-tuning, all LLMs show performance improvements compared to zero-shot inference. Compared to other LLMs, ChatGLM3-6B still underperforms in LoRA fine-tuning, likely due to its weaker ability to align with the target domain of sequence labeling. As a result, the synthesized data produced by ChatGLM3-6B contains a considerable amount of irrelevant information. With the 250-sample setting, Qwen2.5-14B-Instruct demonstrates outperforms other LLMs on most datasets, suggesting that models with larger parameters tend to show enhanced initial performance in low-resource contexts. Nonetheless, with the sample size increases, the performance improvements of Qwen2.5-14B-Instruct on datasets such as Weibo, Taobao, and MSR fell short compared to GLM4-9B-Chat. This could be due to variations in model performance when applied to different domain-specific data distributions. It is important to highlight that the performance of LLMs on most datasets was still below that of conventional sequence labeling baselines, including the the relatively simple BERT-CRF. The findings from the main results indicate that the knowledge contained in LLMs can significantly improve sequence labeling performance. However, they lack the necessary expertise and alignment capabilities for handling domain-specific datasets. Due to biases in domain data distribution, LLMs struggle to identify target entities in particular fields as efficiently as conventional sequence labeling techniques.

Considering the trade-offs between cost and performance, the data augmentation approach leveraging LLMs proposed in this study offers a more practical and efficient solution. This method bridges the gap between LLMs' general knowledge and the specialized requirements of domain-specific sequence labeling tasks.

B Results on Full Datasets

To evaluate the effectiveness of Label Extension Annotation at the full data scale, we conducted additional experiments comparing our method with W^{2}NER, CNN Nested NER, and DiFiNet. The results are shown in Table 6. While performance improvements become less pronounced on certain datasets (e.g., Weibo, Youku, and Resume) compared to the 1000-sample setting, KnowFREE and KnowFREE-F (Deepseek-V3) still outperform all baseline methods on the full datasets, demonstrating their robustness even under high-resource conditions. Although the impact of label extension annotation diminishes as the dataset size increases, it consistently offers improvements over the vanilla KnowFREE, confirming its continued utility. As for enriched explanation synthesis, one of our main motivations was to investigate its performance boundary in low-resource settings. Our analysis shows that its benefits significantly decrease as the sample size grows, with little gain beyond the 500-sample mark. Thus, we can reasonably conclude that enriched explanation synthesis provides limited added value in full-data scenarios.

C Visual Analysis of Enriched Explanation Synthesis

In this section, we use the BERT-based embedding model, text2vec (Ming, 2022), to generate sentence embeddings for the original training and test samples, as well as the enriched explanation samples from the Weibo, Youku, Taobao, Resume, and CMeEE-v2 datasets. These embeddings are projected into a two-dimensional space using the t-SNE algorithm, with the results shown in Fig-

Model	Weibo	Youku	Taobao	Resume	CMeEE	PKU	MSR	UD	CoNLL'03	MIT-Movie
W^{2}NER	72.59	83.62	88.27	96.88	72.97	95.51	97.72	95.00	91.71	74.62
CNN Nested NER	72.31	83.79	88.86	96.67	73.83	93.75	97.69	94.88	91.16	74.86
DiFiNet	73.33	83.69	88.19	96.59	72.29	94.86	97.28	94.85	90.72	74.49
KnowFREE	73.87	84.52	88.97	96.82	73.92	96.59	97.72	95.93	**92.28**	75.32
KnowFREE-F (Deepseek-V3)	**74.15**	**84.57**	**89.12**	**96.93**	**73.95**	**96.67**	**97.76**	**96.02**	92.27	**75.38**

Table 6: Results of sequence labeling datasets on full datasets. The **bold** values indicate the best performance.

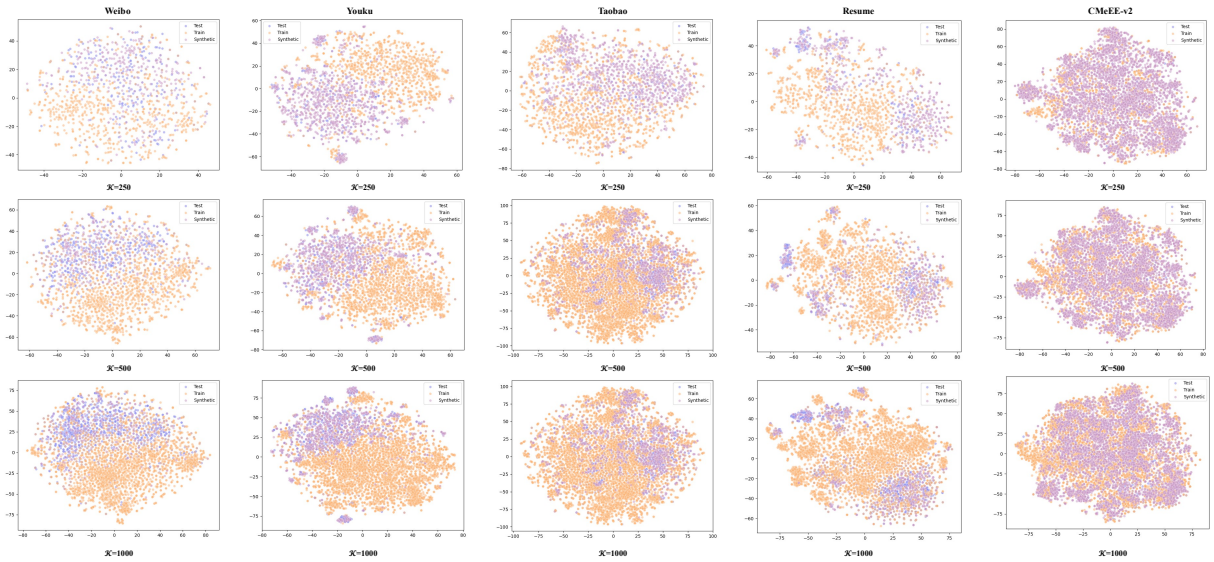

Figure 5: t-SNE visualization of the training, test and enriched explanation samples under different sampling sizes. The synthetic enriched explanation samples are generated by ChatGLM3-6B, and they are represented by the "Synthetic" in the legend.

ure 5. At 250, the sparse distribution of training samples in datasets such as Weibo, Youku, Taobao, and Resume fails to fully cover the semantic space of the test sets, the limitation is observed across all datasets. However, the synthesized samples effectively bridge these gaps in the semantic space, leading to significant performance improvements in low-resource scenarios. As $\mathcal{K}$ increases, the training samples begin to provide more comprehensive coverage of the semantic space for most datasets. At 1000, however, semantic distribution discrepancies are noticeable between some training and synthesized samples in datasets like Youku, Taobao, and CMeEE-v2, potentially introducing noise that may adversely affect model performance. On the Weibo dataset, certain regions of the test set's semantic space remain underrepresented by the original training samples, even at 1000. This explains why models trained with synthesized samples continue to exhibit notable performance advantages over those trained without synthesized samples at this sample size.

These observations further underscore the effectiveness of the enriched explanation synthesis method in improving model performance under low-resource conditions. However, they also highlight that as the sample size increases, the potential adverse effects of synthesized samples, such as semantic noise, become more pronounced.

D Impact of the Number of Heads in Local Attention

To analyze the impact of different numbers of attention heads on model performance, we conducted experiments on four flat NER datasets: Weibo, Youku, Taobao, and Resume, using a sampling size of 1000. The model was trained on the sampled data with extension labels extracted by GLM4-9B-Chat, and the results are presented in Figure 7.

The results suggest that the number of attention heads has a relatively moderate influence on model performance. Notably, increasing the number of heads from 8 to 10 yields the most substantial improvement. Moreover, how feature vectors are distributed across heads proves to be a critical factor. To maintain compatibility as the number of

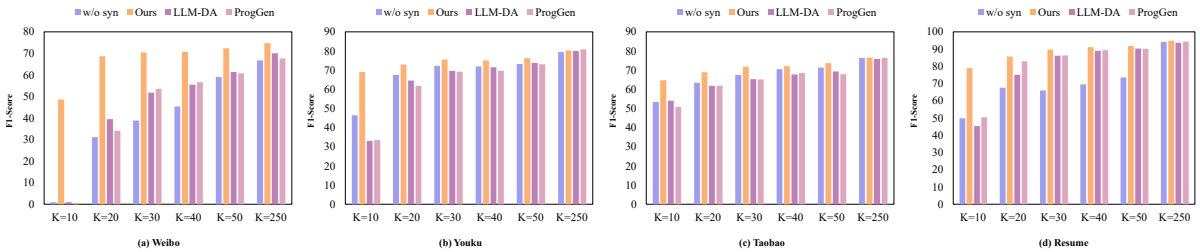

Figure 6: Performance comparison between different data synthesis strategies under k-shot sampling.

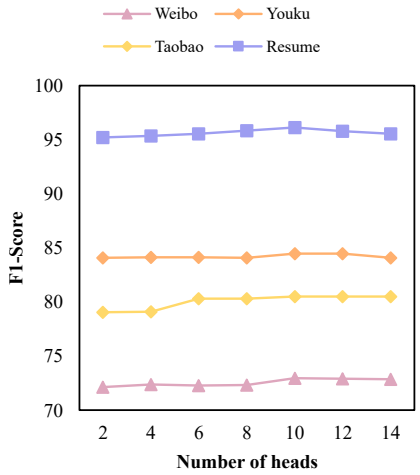

Figure 7: Performance variation with different numbers of heads in the local attention module.

heads increases, we adjusted the feature size to ensure it remains divisible by the number of heads. However, this adjustment did not result in further performance gains and significantly increased computational overhead. Therefore, we adopt ten attention heads in this study as a trade-off between performance and efficiency.

E Comparison of different data synthesis strategies

To evaluate the effectiveness of the enriched explanation synthesis strategy, we reproduced and compared two LLM-based data synthesis methods designed for sequence labeling tasks: LLM-DA and ProgGen. These methods were used to synthesize data from k-shot samples of the original datasets. For consistency, all synthesized samples were annotated using the KnowFREE model trained on the original data without synthesized samples. We conducted experiments on the Weibo, Youku, Taobao, and Resume datasets, all results are presented in Figure 6. The results show that the performance gains from all data synthesis strategies decrease as the number of samples increases. Notably, in scenarios with k $\leq$ 30 on the Youku and Taobao datasets, both LLM-DA and ProgGen lead to performance degradation compared to models trained without synthesized data. This suggests that the synthesized samples generated by these methods may contain inherent semantic distribution biases, which diminish their effectiveness in enhancing performance in certain low-resource domains.

In contrast, our method consistently delivers significantly better performance improvements for k $\leq$ 50, with particularly notable gains on the Weibo dataset. These results demonstrate that the samples synthesized by our approach are more closely aligned with the target domain's distribution and exhibit superior robustness.

F Visualization of the Logits with Extension Labels

To further investigate the interaction between the introduced extension labels and target labels in the model, we visualized the logit scores of extension labels corresponding to each predicted target label position in the test set. These scores were aggregated by summing and averaging across label categories, and the results are displayed as a heatmap in Figure 8. In each heatmap subplot, the horizontal axis represents the target labels, while the vertical axis corresponds to the extension labels.

The results reveal that certain extension labels exhibit strong correlations with specific target labels. For instance, in the Weibo dataset, both "PER.NAM" and "PER.NOM" show a notable association with the extension label "PERSON". Similarly, in the Resume dataset, "COUNT" demonstrates strong correlations with the extension labels "Country," "Location," and "description". Incorporating these significant relationships during training allows the model to leverage co-occurrence patterns, enhancing its ability to perform fine-grained semantic understanding and improving target entity

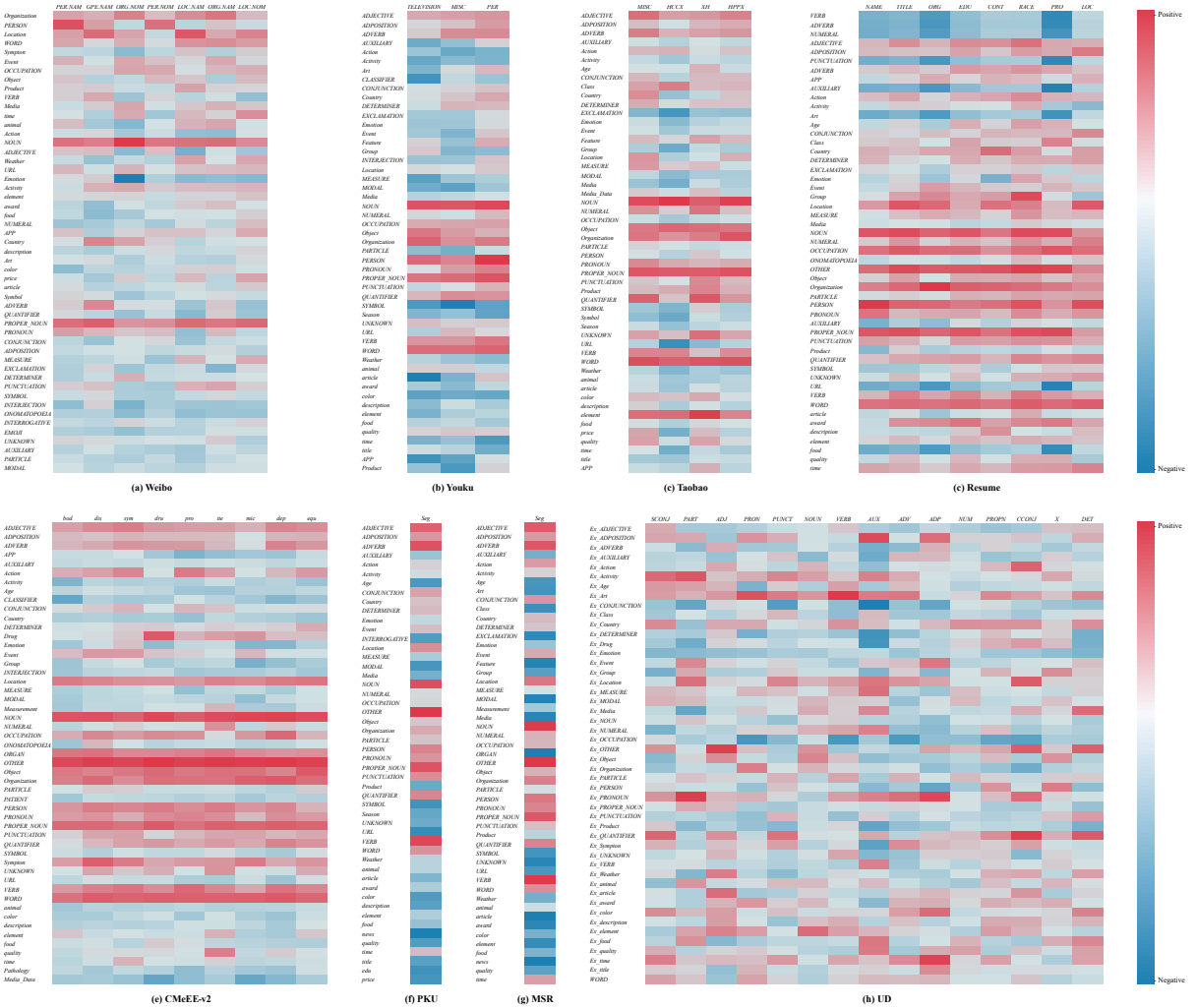

Figure 8: Heatmap visualization of logits scores between target labels and extension labels on the test sets.

prediction.

However, some extension labels were observed to have strong correlations with all target labels. Since the KnowFREE model produces independent probability scores for each target label, the influence of such extension labels on target entity predictions is generally limited when their number is small, as their training weights are relatively low. Conversely, when these extension labels become overly numerous, they can negatively impact model training. In such cases, reducing their weights further can help mitigate these effects and enhance overall model performance.

G Statistics of Datasets

The detailed statistics of the datasets are shown in Table 7. These datasets span various domains, including Social Media, E-commerce, and Medical, enabling a comprehensive evaluation of the model's performance across different fields.

Dataset	Dev	Test	Label Types	Domain
Weibo	0.27k	0.27k	8	Social Media
Youku	1.00k	1.00k	3	Video Content
Taobao	1.00k	1.00k	4	E-commerce
Resume	0.46k	0.48k	8	Human Resources
CMeEE-v2	4.98k	4.98k	9	Medical
PKU	1.00k	2.04k	1	News
MSR	1.00k	3.99k	1	News
UD	0.50k	0.50k	16	News, Literature
CoNLL'03	3.47k	3.68k	4	News
MIT-Movie	1.00k	1.95k	12	Entertainment

Table 7: Statistics of development sets, test sets, label types and domains of all datasets.

H More Experiment Settings

In this section, we describe the additional experimental parameter settings for our method. In the pipeline of label extension annotation, the pre-trained embedding model of $\mathcal{M}$ is set as "text2vec" (Ming, 2022), the Top-p value is set as five, and the threshold ϵ is set as 1.5. In the training stage, the

hidden size D, D', and $\tilde{D}$ are set as 768, 200, and 200, respectively. The activation function of σ and σ^* are defined as Leaky ReLU and GeLU, respectively. In the local multi-head attention module, the window size ω is set as three and the number of attention heads K is set as ten. In the sequence labeling model, we distinguish between the learning rate for the PLM and other modules, setting them to 2e-5 and 1e-3, respectively. For the weight α of extension labels, we introduce a dynamic weight calculation mechanism to handle the influence of frequently occurring extension labels (e.g., POS tags). These frequent labels can affect the gradient calculation, leading to reduced attention to target labels. To address this, we calculate the count C_i of each extension label and the average count $\hat{C}$ of target labels, and then compute the weight coefficient α_i as follows:

$$\alpha_i = 0.5 \times (\hat{C}/C_i). \tag{15}$$

Dataset	Weight Decay	β	Type
Weibo	1e-3	1.0	Flat NER
Youku	1e-2	0.4	Flat NER
Taobao	1e-3	0.4	Flat NER
Resume	1e-3	0.4	Flat NER
CMeEEv2	1e-3	0.4	Nested NER
PKU	1e-2	1.0	Word Segment
MSR	1e-2	1.0	Word Segment
UD	1e-2	1.0	POS Tagging
CoNLL'03	1e-3	0.4	Flat NER
MIT-Movie	1e-2	0.4	Flat NER

Table 8: Settings of β and weight decay across different datasets.

The training weight for synthesized samples (β) and the weight decay parameter are provided in Table 8. As shown, for most NER datasets with more complex entity semantics, we use a smaller weight decay parameter to improve model fitting during training. In contrast, for POS tagging and tokenization datasets, we apply a larger weight decay to prevent overfitting. Additionally, for NER datasets, where entity labels are more prone to noise from synthesized samples, we set $\beta = 0.4$. On the other hand, for datasets with strong baseline performance, increasing β to 1.0 helps the model better utilize synthesized samples during training. Our implementation is built on the Huggingface Transformers (Wolf et al., 2020), and all experiments are conducted using two NVIDIA A6000 GPUs for both training and inference.

I Prompts

In this section, we present detailed examples of our workflow prompts for label extension annotation in Figure 9, 10 and enriched explanation synthesis in Figure 11, 12. Since the target dataset is entirely in Chinese, all original prompts are written in Chinese. The English portions in the prompt examples are translations of the original prompts.

Entity Extraction Prompt

指令:请识别并抽取输入句子的命名实体 ，并使用JSON格式的数组进行返回，子项包括entity和type属性：
条件： 1. 输出格式为[{entity: '', type: ''}],其中entity表示所提取的实体文本, type表示所提取的实体类型, 一个entity对应一个type
2. 如果不存在任何实体，请输出空数组[]

Instruction: Please identify and extract the named entities from the input sentence and return them in a JSON array format. Each item should include the attributes `entity` and `type`:

Conditions: 1. The output format should be [{entity: '', type: ''}], where `entity` represents the extracted entity text, and `type` represents the extracted entity type. Each entity corresponds to one type.
2. If no entities exist, output an empty array [].

Figure 9: The entity extraction prompt in label extension annotation.

Entity Explanation Prompt

指令: 你作为拥有丰富知识储备的专家，需要根据我给出的样例进行续写，给学生们解释样例中包含的实体含义，我会给定你样例和其包含的实体加实体类型，请续写样例的后文，并解释每个实体在样例中的含义。
样例： 现任中国科技大学商学院院长 、中国现场统计学会副理事长、美国当代统计索引CIS通讯编辑，为美国ASA、IMS会员。
实体： ['中国科技大学商学院(组织机构)', '院长(头衔职称)'...]

Instruction: As an expert with extensive knowledge, you are required to continue writing based on the provided sample and explain the entities included in the sample for students. I will give you a sample along with the entities it contains and their corresponding entity types. Please continue writing the sample and explain what each entity means in the sample.

Sample: Currently the Dean of the School of Management at the University of Science and Technology of China, Vice Chairman of the Chinese Society of Probability and Statistics, Communications Editor of the Contemporary Index of Statistics (CIS) in the United States, and a member of the American Statistical Association (ASA) and the Institute of Mathematical Statistics (IMS).

Entities: ['School of Management at the University of Science and Technology of China (Organization)', 'Dean (Title)' ...]

Figure 11: The entity explanation prompt in enriched explanation synthesis.

POS Tag Extraction Prompt

指令:请提取输入句子的词性 (POS)，并使用JSON格式的数组进行返回，子项包括word和pos属性：
条件： 1. 输出格式为[{word: '', pos: ''}],其中word表示所提取的文本, pos表示所提取的词性, 一个word对应一个pos
2. 请务必将输入中**所有字符**和**标点**都进行标注

Instruction: Please extract the Part-of-Speech (POS) of the input sentence and return them in a JSON array format. Each item should include the attributes `word` and `pos`:

Conditions: 1. The output format should be [{word: '', pos: ''}], where `word` represents the extracted text, and `pos` represents the corresponding part-of-speech. Each word corresponds to one `pos`.
2. Ensure that all **characters** and **punctuation** marks in the input are annotated.

Figure 10: The POS tag extraction prompt in label extension annotation.

Extension Description Prompt

指令: 你作为拥有丰富知识储备的专家，需要将我给出的样例中包含的"关键短语"(如**实体**或**序列**)进行抽取，并给学生们解释这些关键短语在样例中的含义，我会给定你样 例，请先将"关键短语"抽取出来，其次再根据"关键短语"续写样例的后文，并解释每个"关键短语"在样例中的含义。
样例： 现任中国科技大学商学院院长 、中国现场统计学会副理事长、美国当代统计索引CIS通讯编辑，为美国ASA、IMS会员。

Instruction: As an expert with extensive knowledge, you are required to extract "key phrases" (such as **entities** or **phrases**) from the given sample and explain their meaning in the context of the sample to students. I will provide you with a sample. Next, you should extract the "key phrases," then continuous generate the sample's content based on these "key phrases" and explain the meaning of each "key phrase" in the sample.

Sample: Currently the Dean of the School of Management at the University of Science and Technology of China, Vice Chairman of the Chinese Society of Probability and Statistics, Communications Editor of the Contemporary Index of Statistics (CIS) in the United States, and a member of the American Statistical Association (ASA) and the Institute of Mathematical Statistics (IMS).

Figure 12: The extension description prompt in enriched explanation synthesis.