# Character-Boundary-Aware and Type-Decoupled Named Entity Recognition Enhanced by a Post-Correction Model

**Yongshen Zhu[1], Chengjie Sun[1,2*], Conghui Zhu[1], Bingquan Liu[1], Lei Lin[1]**

[1]Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China
[2]National Research Center for Language Technology and Digital Economy, Harbin Institute of Technology, Harbin 150001, China

**Keywords:** NER; LLM; Character-boundary-aware; Post-correction

Citation: Zhu Y.S., Sun C.J., Zhu C.H., et al.: Character-Boundary-Aware and Type-Decoupled Named Entity Recognition Enhanced by a Post-Correction Model. Data Intelligence, 2026, Vol. 8, Art. No.: 20250095. DOI: https://doi.org/10.3724/2096-7004.di.2025.0095

## ABSTRACT

Named Entity Recognition (NER) aims to identify entities with specific semantic meanings from text and classify them into predefined categories. With the rapid progress of generative large language models (LLMs), their strong capabilities in text comprehension and generation have sparked growing interest in applying them to information extraction tasks, including NER, within a generative paradigm. However, LLMs are often regarded as "black-box" systems, making it difficult to interpret the reasoning behind their predictions in entity recognition. In contrast, traditional BERT-based models explicitly assign labels to each token, ensuring precise capture of entity boundaries, while LLMs tend to infer spans from context, which may weaken their sensitivity to boundary information. To overcome these challenges, we propose a novel framework: CBATD-PC. Our approach integrates an instruction-tuning strategy for model construction and introduces a character-level boundary labeling mechanism. By decoupling the NER process into two stages, boundary detection and type classification, we establish a more structured and interpretable prediction pipeline. Moreover, by leveraging the contextual reasoning strengths of LLMs, we design a post-correction module that fine-tunes and refines the initial extraction results, thereby improving the overall accuracy and robustness of entity recognition.

*Corresponding author: Chengjie Sun (E-mail: sunchengjie@hit.edu.cn; ORCID: 0000-0001-9081-1410).

© 2026 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

## 1. INTRODUCTION

Named Entity Recognition (NER) is a fundamental task in natural language processing (NLP), with the aim of identifying entities with specific semantic meanings, such as person names, organization names, and location names from unstructured text, and to classify them into predefined categories. Traditional NER approaches, particularly those based on BERT [1], have achieved remarkable success by performing explicit token-level annotation within sequence labeling frameworks, thereby accurately capturing entity boundaries. However, BERT-based models often exhibit limited generalization across different domains. With the rapid advancement of large language models (LLMs), researchers have increasingly explored leveraging their strong contextual understanding and generative capabilities to perform NER under a generative paradigm [2], with the goal of developing a unified extraction model capable of generalizing across diverse domains.

In recent years, researchers have fine-tuned open-source large language models (LLMs) on diverse named entity recognition datasets, significantly enhancing their adaptability across different domains. To tackle cross-domain entity extraction tasks, various approaches have been proposed. For instance, UIE [3] pioneered the use of LLMs for generative unified modeling by introducing a text-to-structure generation framework for information extraction. InstructUIE [4] proposed a unified information extraction framework based on instruction tuning, guiding LLMs through natural language instructions to perform extraction tasks. ChatUIE [5] combined reinforcement learning with pre-trained supervised fine-tuning and reward models to coordinate multiple information extraction tasks. Yayi-UIE [6] introduced an end-to-end chat-enhanced instruction tuning framework for general information extraction, leveraging both dialogue and extraction datasets to improve performance. Finally, KnowCoder [7] employed formal programming languages to uniformly represent structured knowledge, integrating symbolic knowledge graphs with neural LLMs, thereby enhancing both the precision and generalization capability of knowledge extraction.

However, most of these methods adopt an end-to-end approach, directly generating outputs based on instructions without explicitly modeling intermediate steps. This makes it challenging to trace the reasoning process behind specific predictions in entity recognition. Furthermore, when training large models with cross-entropy loss, each token contributes equally to the loss function. Consequently, the model may prioritize capturing the overall span rather than focusing on critical boundary information, which can impair its ability to accurately detect entity boundaries and ultimately result in less precise extractions.

To address these challenges, we propose a model framework based on instruction tuning and introduce a character-level entity/non-entity boundary tagging mechanism, enabling a stage-wise decoupling of the entity recognition process. By splitting entity recognition into two distinct stages—boundary detection and type classification—we not only improve the accuracy of entity extraction but also enhance the interpretability of the model. Furthermore, we design a cascade-style, fine-tuned language model framework for entity extraction and post-correction, allowing iterative optimization and refinement of the extraction results.

We conducted experiments on the MSRA NER dataset [8] using the generative large language models Baichuan2-13B [9] and Qwen-2.5-14B-Instruct [10]. Our method, CBATD-Baichuan, outperforms Yayi-UIE, which was also trained on the Baichuan model. Moreover, CBATD-PC-Qwen achieves performance slightly surpassing the current state-of-the-art (SOTA) results.

## 2. RELATED WORK

### *2.1 Generative Large Language Model*

In recent years, Generative Large Language Models (GLLMs) have made remarkable progress in the field of natural language processing. Vaswani et al. [11] introduced the Transformer architecture, which revolutionized language model design by relying entirely on the self-attention mechanism. This architecture enables efficient modeling of global dependencies within input sequences. Building on this foundation, Google proposed BERT, which utilizes bidirectional context to achieve breakthrough performance across a wide range of natural language understanding tasks.

At the same time, OpenAI introduced the GPT series [12], focusing on advancing text generation capabilities. The GPT models adopt a unidirectional decoder architecture and, after pre-training on large-scale corpora, can perform downstream tasks with minimal or even no task-specific examples—ushering in a new era for generative language models.

As computational resources expanded, researchers began exploring even larger models. Studies have shown that model performance generally improves with an increased number of parameters. For example, GPT-3 [13], with over 175 billion parameters, demonstrated exceptional generalization and few-shot learning capabilities. Subsequently, Meta released the LLaMA series [14], significantly advancing research on large language models within the open-source community.

Most generative large language models follow a two-stage training paradigm: first, unsupervised pre-training on large-scale corpora, followed by supervised fine-tuning on task-specific datasets. In addition, techniques such as Prompt Engineering [15] and In-Context Learning [16] have emerged as effective strategies to guide model outputs without modifying the model's parameters.

### *2.2 Named Entity Recognition*

Named Entity Recognition (NER) is a fundamental task in natural language processing, aimed at identifying and classifying entities with specific semantic categories from text, such as person names (PER), locations (LOC), and organization names (ORG). In recent years, driven by the rapid development of deep learning and pre-trained language models, NER methodologies have continuously evolved—from early rule-based and feature-engineering approaches to modern architectures powered by large pre-trained models.

GPT-NER [17] leverages large language models for NER by converting the sequence labeling task into a text generation task. Yayi-UIE introduces an end-to-end, chat-enhanced instruction tuning framework

for general information extraction, utilizing both dialogue data and information extraction datasets to improve performance. However, these approaches often struggle to trace the reasoning behind specific entity predictions. In addition, such models tend to focus on overall entity spans rather than on critical boundary information.

In contrast, our method introduces a character-level boundary tagging mechanism, decoupling the entity recognition process into two distinct stages: boundary detection and type classification. This stage-wise decoupling improves both the interpretability and the accuracy of entity recognition.

Additionally, Double-Checker [18] employs LLMs as "checkers" to validate and refine predictions in few-shot NER tasks, thereby improving model performance in low-resource settings. DEER [19] introduces a novel Label-Guided In-Context Learning (ICL) framework, which enhances in-context learning effectiveness through label statistics and error reflection mechanisms. It achieves near-supervised performance in few-shot scenarios without any fine-tuning, relying solely on prompt engineering.

The aforementioned LLM-based verification approaches rely primarily on prompt-based inference. In contrast, our proposed method improves upon this paradigm by fine-tuning a large model to perform secondary analysis and correction of the initial extraction results, thereby further enhancing the overall extraction performance.

## 3. METHODOLOGY

Figure 1 presents an overview of the CBATD-PC framework. The first stage is handled by the primary model, CBATD-NER, which performs initial entity recognition. This stage incorporates a character-level

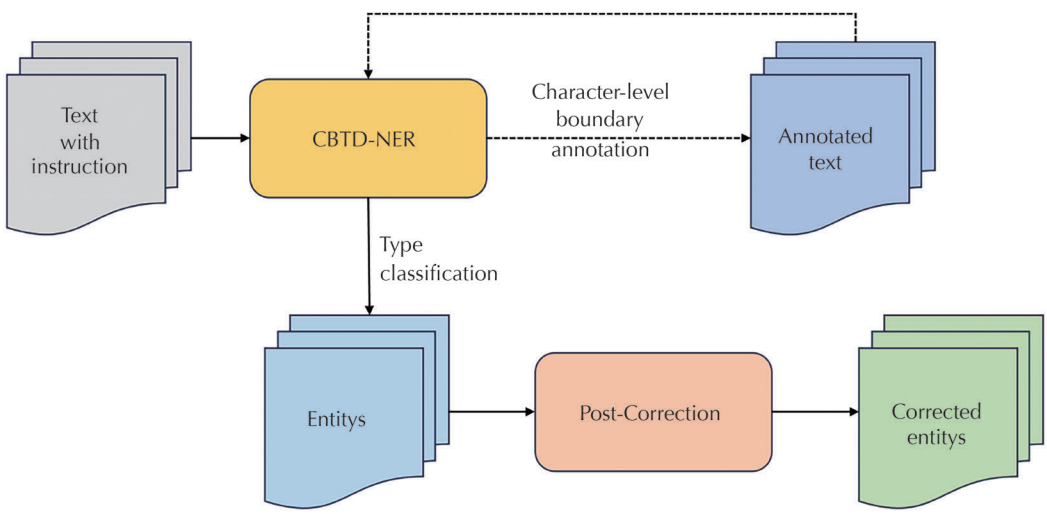



**Figure 1.** The overview of the CBATD-PC Framework.

boundary annotation mechanism, decomposing the entity recognition process into two distinct phases: boundary detection and type classification, thereby achieving a stage-wise decoupling of the recognition process. The model is built using an instruction-tuning strategy. In the second stage, a refinement model, Post-Correction, is introduced to further enhance the accuracy of entity recognition by refining the outputs of the primary model. This correction model is also constructed using instruction tuning.

### *3.1 Character-Boundary-Aware and Type-Decoupled*

Specifically, we decompose the entity recognition task into two distinct stages: character-level entity/non-entity boundary detection and entity type classification, thereby constructing a hierarchical entity extraction framework. The core idea is to employ a character-level annotation mechanism to accurately identify the start, middle, and end positions of entities, as well as the distribution of all non-entity characters. Based on the extracted entity and non-entity segments, the model then performs type classification to generate the final extraction results.

#### *3.1.1 Boundary Annotation Mechanism*

Considering that the text primarily consists of Chinese or English characters, and to facilitate the observation of annotation results, we adopt a five-category character-level annotation scheme (1/2/3/4/0), defined as follows:

1 (B) : Beginning of an entity.

2 (I) : Inside an entity.

3 (E) : End of an entity.

4 (S) : Single-character entity.

0 (O) : Non-entity (outside) characters.

#### *3.1.2 Two-Stage Modeling Process*

The proposed entity extraction framework consists of the following two stages:

Stage 1: Entity/Non-entity Boundary Detection

Input: An original character sequence $X = [x_1, x_2, \ldots, x_n]$.

Output: A boundary-annotated sequence $Y_{\text{bound}} = [x_1 y_1^b, x_2 y_2^b, \ldots, x_n y_n^b]$, where $y_i^b \in \{0, 1, 2, 3, 4\}$ denotes the boundary tag for the $i$-th character.

Stage 2: Entity Type Classification

Input: The boundary-annotated sequence $Y_{\text{bound}} = [x_1 y_1^b, x_2 y_2^b, \ldots, x_n y_n^b]$ generated in Stage 1.

Output: The final extraction result with entity types and position markers $Y_{\text{type}} = [y_1^t, y_2^t, \ldots, y_m^t]$, where $y_i^t \in T$, and $T$ represents the set of entity type labels combined with position information.

### *3.1.3 Chain-of-Thought Based Two-Stage Integration Mechanism*

To effectively integrate the two-stage entity recognition process, we design a stage-aware prompting strategy leveraging the autoregressive nature of large language models (LLMs).

Since LLMs generate outputs sequentially, we structure the prompt to guide the model through a step-by-step reasoning process: first performing character-level boundary annotation, followed by entity type classification and final result generation. This approach emulates a chain-of-thought reasoning pattern, enabling the model to decompose complex tasks into structured subtasks while maintaining end-to-end inference efficiency.

As shown in Figure 2, the input format designed in this study consists of two main components: text input and guidelines. Upon receiving the input, the model is instructed to follow a specified output format: first performing boundary annotation on the text, then determining entity types based on the generated annotations, and finally producing structured entity extraction results.

In summary, the proposed method offers the following key advantages:

(i) **Incorporation of Entity Context, Including Non-entity Spans**: Each character's label depends not only on its own semantics but also on the contextual information and labels of preceding characters. By integrating contextual information from both inside and outside entities during training, the model gains a stronger ability to recognize entities. Furthermore, all non-entity spans in the text are included as part of the training data. Unlike traditional generative information extraction methods that directly produce final results, our approach enables the model to understand entity spans based on context and per-character annotations. This includes identifying both entity and non-entity content, providing a solid foundation for accurate entity extraction.

(ii) **Enhanced Boundary Detection Capability**: Our method employs a boundary annotation scheme (1/2/3/4/0), explicitly distinguishing the start, middle, and end positions of entities, as well as non-entity characters. Compared to conventional generative approaches, this labeling strategy significantly enhances the model's ability to accurately delineate entity boundaries.

(iii) **Alignment with Chain-of-Thought Reasoning**: The method follows a two-step reasoning process: first, boundary detection is performed by annotating each character using the 1/2/3/4/0 tagging scheme, effectively identifying potential entity and non-entity spans; then, type classification is conducted based on these identified spans. This stepwise process emulates chain-of-thought reasoning, allowing the model to first learn boundary features and subsequently type features. As a result, the model develops a clearer understanding of the task structure, reduces errors caused by leapfrog reasoning, enhances interpretability, and ultimately improves entity recognition accuracy.

**A Sample Prompt for NER Instruction Tuning**

**Guidelines:**

“【实体抽取】抽取文本中可能存在的实体，请先于文本中在每一个字符后将你认为的实体左边界标为 1，实体中间的部分标为 2，实体右边界标为 3，如果实体为单字实体则标注为 4，其余标注为 0，最后根据标注的文本信息顺序生成结果。并以 json{“anno_text”: “ 标注文本 ”，“entity”: [{“type”: “ 人物 / 组织 / 地点 ”，“text”: “ 实体 span”, “offset”: [ 实体头 index, 实体尾 index]}]} 格式输出。”

“【Entity Extraction】Extract possible entities in the text. Please first annotate each character in the text as follows: mark the left boundary of an entity as 1，the middle part of an entity as 2，the right boundary of an entity as 3. If the entity consists of a single character, mark it as 4. All other positions should be marked as 0. Finally, generate the result based on the annotated text in order. The output format should be in JSON:{ “anno_text”: “Annotated text”, “entity”: [ {“type”: “Person/Organization/Location”, “text”: “Entity span”, “offset”: [Start index, End index]}]}

**Input (“text: “+ text + guidelines):**

“ 文本：”+“ 国正先生在我心中就是这样的一位学长。”+ “\n” + guidelines,

**Output:**

```
“{
  “anno_text”: “ 国 1 正 3 先 0 生 0 在 0 我 0 心 0 中 0 就 0 是 0 这 0 样 0 的 0—0 位 0 学 0 长 0。”,
  “entity”: [
   {
     “type”: ＂人物”，
     “text”: ＂国正”，
     “offset”: [0, 1]
   }
]
}”
```

**Figure 2.** A Sample of NER Training Data.

### *3.2 Post-Correction*

Based on the output of the primary model (CBATD-NER) from the first stage, we introduce a refinement model in the second stage to further correct and enhance the results produced by the primary model.

When the primary model achieves nearly 100% F1 score on the training set, the remaining errors—such as false positives and missed entities—represent the most challenging cases in entity recognition. Since the primary model already possesses a strong foundational ability for entity detection, the primary role of the refinement model is to learn how to correct these difficult error patterns.

To this end, we select a subset of data for instruction tuning, guiding the refinement model to acquire the capability to identify and correct such errors. The prompt format and training data structure used for fine-tuning are illustrated in Figure 3.

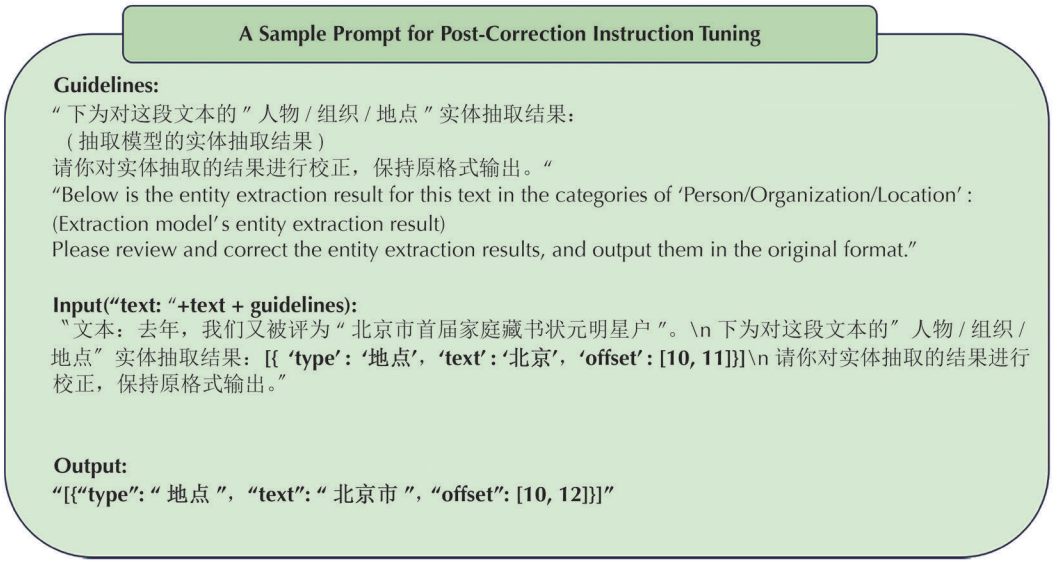



**Figure 3.** A Sample of Post-Correction Training Data.

## 4. EXPERIMENTS AND ANALYSIS

### *4.1 Datasets and Experimental Setup*

#### *4.1.1 Datasets*

We have conducted experimental validation on the MSRA entity extraction dataset, comparing several existing entity extraction methods to analyze their performance differences. The MSRA dataset is a benchmark for Chinese named entity recognition in the news domain, consisting of a training set with 46,364 instances and a test set with 4,365 instances. The entity types include locations (LOC), person names (NAME), and organization names (ORG).

#### *4.1.2 Implementation Details*

For CBATD-NER, we followed the instruction-tuning prompt template described previously to preprocess the MSRA dataset and constructed a fine-tuning dataset suitable for large language models. Subsequently, we performed instruction tuning based on two base models: Qwen2.5-14B-Instruct and Baichuan2-13B. The evaluation metric used was the entity-level F1 score.

For the Post-Correction model, we selected certain early checkpoints of CBATD-NER during training and used them to make predictions on the MSRA training set. From these predictions, we chose a subset achieving an F1 score of 91.65, forming our candidate data source for refinement training—referred to as the “correction fine-tuning data source.” In the first approach, we extracted all samples from this data

source that were incorrectly recognized by CBATD-NER to construct a specialized fine-tuning dataset aimed at learning error correction patterns. In the second approach, we included all predicted samples from the correction fine-tuning data source in the training set—both those incorrectly recognized by the primary model and those correctly recognized that required no modification.

### *4.2 Overall Comparison Results*

We evaluate **CBATD-NER** on the MSRA dataset and compare it with previously reported baselines. In addition, two ablation studies are conducted to analyse the influence of the backbone choice and the proposed post-correction strategy.

#### *4.2.1 Overall comparison*

Table 1 summarises the overall results. The vanilla CBATD-NER built upon Qwen-2.5-14B-Instruct reaches an F1 of 96.44, which is already competitive with the current state-of-the-art (SOTA) BERT-MRC + DSC (96.72) [20]. After applying the post-correction module, the performance further improves to **96.76**, establishing a new SOTA while maintaining a simple pipeline.

**Table 1.** Performance comparison with previous methods on the MSRA dataset.

| Model | Backbone | MSRA (F1) |
|---|---|---|
| Yayi-UIE | Baichuan2-13B | 95.57 |
| SENCR (AAAI'24) | BERT-base | 96.17 |
| BERT-MRC+DSC (ACL'20) | BERT-MRC | 96.72 |
| CBATD-NER (Qwen) | Qwen-2.5-14B-Instruct | 96.44 |
| CBATD-NER + Post-Correction | Qwen-2.5-14B-Instruct | **96.76** |

#### *4.2.2 Ablation study*

The ablation results in Table 2 provide two key insights.

(i) **Backbone selection.** Replacing Qwen with Baichuan2-13B degrades the score slightly to 96.11, yet still surpasses Yayi-UIE (95.57) built on the same backbone. This confirms that the performance gains of CBATD-NER arise from its two-stage architecture and global token discrimination design, rather than from the backbone alone.

(ii) **Post-correction strategy.** Training the post-correction model only on error samples ("*error-only training*" in Table 2) drastically reduces performance to 75.59, as the model tends to over-correct and alter originally correct predictions. When all predictions are included in the correction fine-tuning (last row of Table 2), the model learns to distinguish genuine errors, achieving the best F1 score of 96.76.

**Table 2.** Ablation study on the MSRA dataset.

| Model Variant | Backbone | MSRA (F1) |
|---|---|---|
| Yayi-UIE | Baichuan2-13B | 95.57 |
| CBATD-NER (Baichuan) | Baichuan2-13B | 96.11 |
| CBATD-NER (Qwen) | Qwen-2.5-14B-Instruct | 96.44 |
| CBATD-NER + Post-Correction (error-only training) | Qwen-2.5-14B-Instruct | 75.59 |
| CBATD-NER + Post-Correction (full training) | Qwen-2.5-14B-Instruct | **96.76** |

These results demonstrate that (i) CBATD-NER is robust across different backbones, and (ii) the proposed two-stage correction mechanism is essential for achieving state-of-the-art performance.

#### *4.2.3 Case study*

To illustrate our method's qualitative benefits, Table 3 presents a real example from the MSRA test set. The vanilla CBATD-NER mistakenly splits the long organisational name '中国致公党十一大' into two consecutive entities, while our post-correction module merges them into the correct span and thereby matches the gold annotation.

**Table 3.** Qualitative comparison between CBATD-NER variants on the sentence "中共中央致中国致公党十一大的贺词".

| System | Extracted entities (type, text, offset) |
|---|---|
| CBATD-NER | ['type': '组织', 'text': '中共中央', 'offset': [0, 3], 'type': '组织', 'text': '中国致公党', 'offset': [5, 9], 'type': '组织', 'text': '十一大', 'offset': [10, 12]] |
| Post-Correction | ['type': '组织', 'text': '中共中央', 'offset': [0, 3], 'type': '组织', 'text': '中国致公党十一大', 'offset': [5, 12]] |
| Gold reference | ['type': '组织', 'text': '中共中央', 'offset': [0, 3], 'type': '组织', 'text': '中国致公党十一大', 'offset': [5, 12]] |

As can be seen, the post-correction module successfully suppresses the over-segmentation error, underscoring the practical value of the two-stage design beyond mere numerical gains.

## 5. CONCLUSION AND FUTURE WORK

We propose a novel instruction-tuning-based Named Entity Recognition framework, CBATD-NER, which effectively enhances the performance of large language models on NER tasks by introducing a character-level boundary annotation mechanism and a two-stage architecture that separates boundary

detection from type classification. This approach not only improves entity recognition accuracy but also enhances model interpretability. To further boost performance, we introduce a Post-Correction Module that iteratively optimizes and refines the outputs of the base model. Experimental results demonstrate that the corrected outputs outperform both the original CBATD-NER model and the current state-of-the-art BERT-MRC + DSC, fully validating the effectiveness of the proposed two-stage correction mechanism.

It should be noted that the framework relies on large-scale models and instruction-tuning-based fine-tuning, which entails significant computational resources and memory consumption. While the method achieves state-of-the-art performance, deploying and training such models may be challenging in resource-constrained environments. Future work could explore more resource-efficient strategies, such as model distillation or parameter-efficient fine-tuning, to make the approach more accessible.

## AUTHOR CONTRIBUTIONS

Yongshen Zhu: Investigation, Conceptualization, Methodology, Software, Writing – original draft. Chengjie Sun: Resources, Writing – review & editing, Supervision. Conghui Zhu: Resources, Supervision. Bingquan Liu: Writing – review & editing, Supervision. Lei Lin: Methodology, Supervision.

## ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China under grant 2023YFC3804600.

## REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. doi: 10.48550/arXiv.1810.04805.

[2] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen, "Large language models for generative information extraction: A survey," *Frontiers of Computer Science*, vol. 18, no. 6, Art. no. 186357, 2024.

[3] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu, "Unified structure generation for universal information extraction," *arXiv preprint arXiv:2203.12277*, 2022. doi: 10.48550/arXiv.2203.12277.

[4] X. Wang, W. Zhou, C. Zu, H. Xia, T. Chen, Y. Zhang, R. Zheng, J. Ye, Q. Zhang, T. Gui, *et al.*, "InstructUIE: Multi-task instruction tuning for unified information extraction," *arXiv preprint arXiv:2304.08085*, 2023.

[5] J. Xu, M. Sun, Z. Zhang, and J. Zhou, "ChatUIE: Exploring chat-based unified information extraction using large language models," *arXiv preprint arXiv:2403.05132*, 2024.

[6] X. Xiao, Y. Wang, N. Xu, Y. Wang, H. Yang, M. Wang, Y. Luo, L. Wang, W. Mao, and D. Zeng, "YAYI-UIE: A chat-enhanced instruction tuning framework for universal information extraction," *arXiv preprint arXiv:2312.15548*, 2023.

[7] Z. Li, Y. Zeng, Y. Zuo, W. Ren, W. Liu, M. Su, Y. Guo, Y. Liu, X. Li, Z. Hu, *et al.*, "KnowCoder: Coding structured knowledge into LLMs for universal information extraction," *arXiv preprint arXiv:2403.07969*, 2024.

[8] G.-A. Levow, "The third international Chinese language processing bakeoff: Word segmentation and named entity recognition," in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 108–117.

[9] A. Yang, B. Xiao, B. Wang, B. Zhang, C. Bian, C. Yin, C. Lv, D. Pan, D. Wang, D. Yan, *et al*., "Baichuan 2: Open large-scale language models," *arXiv preprint arXiv:2309.10305*, 2023.

[10] Qwen Team, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. doi: 10.48550/arXiv.1706.03762.

[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI Technical Report*, 2018.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al*., "Language models are few-shot learners," A*dvances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al*., "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[15] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," *arXiv preprint arXiv:2402.07927*, 2024.

[16] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, *et al*., "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[17] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, and G. Wang, "GPT-NER: Named entity recognition via large language models," *arXiv preprint arXiv:2304.10428*, 2023.

[18] W. Chen, L. Zhao, Z. Zheng, T. Xu, Y. Wang, and E. Chen, "Double-checker: Large language model as a checker for few-shot named entity recognition," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3172–3181, 2024.

[19] F. Bai, H. Hassanzadeh, A. Saeedi, and M. Dredze, "Label-guided in-context learning for named entity recognition," *arXiv preprint arXiv:2505.23722*, 2025.

[20] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced NLP tasks," *arXiv preprint arXiv:1911.02855*, 2019.