Double-Checker: Large Language Model as a Checker for Few-shot Named Entity Recognition

Wei Chen[1], Lili Zhao[1], Zhi Zheng[1*], Tong Xu[1*], Yang Wang[2], Enhong Chen[1]
[1]University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence, [2]Anhui Conch Information Technology Engineering Co., Ltd.,
{chenweicw, liliz, zhengzhi97}@mail.ustc.edu.cn, tongxu@ustc.edu.cn
wangyang@chinaconch.com, cheneh@ustc.edu.cn

Abstract

Recently, few-shot Named Entity Recognition (NER) has attracted significant attention due to the high cost of obtaining high-quality labeled data. Decomposition-based methods have demonstrated remarkable performance on this task, which initially train a type-independent span detector and subsequently classify the detected spans based on their types. However, this framework has an evident drawback as a domain-agnostic detector cannot ensure the identification of only those entity spans that are specific to the target domain. To address this issue, we propose Double-Checker, which leverages collaboration between Large Language Models (LLMs) and small models. Specifically, we employ LLMs to verify candidate spans predicted by the small model and eliminate any spans that fall outside the scope of the target domain. Extensive experiments validate the effectiveness of our method, consistently yielding improvements over two baseline approaches. Our code is available at github.com/fanshu6hao/Double-Checker.

1 Introduction

In recent years, few-shot Named Entity Recognition (NER) has attracted significant attention due to the high cost of obtaining high-quality labeled data (Ma et al., 2022a; Agrawal et al., 2022). This task mainly focuses on enabling the model to learn from a resource-rich source domain dataset, and further requires the model to predict unseen entity types in a resource-scarce target domain based on a small amount of data, i.e., the support data (Ma et al., 2022a; Das et al., 2022).

To solve the above problem, a common approach is to decompose the task into entity span detection and entity type classification (Chen et al., 2023; Li et al., 2023). Specifically, a type-independent entity span detector is first trained, and then the type classification is performed according to the detection spans. Since the span detector trained in the first stage does not need to focus on specific entity types, it can effectively reduce the distribution gap between the source domain and the target domain, and has excellent performance (Wang et al., 2022; Ma et al., 2022b). However, this paradigm has an obvious drawback: a domain-agnostic detector cannot guarantee that the entity span identified is specific to the target domain, and it will obviously identify many non-target domain candidates [1].

Fortunately, Large Language Models (LLMs) have shown remarkable performance on various natural language processing tasks (Wang et al., 2024), such as semantic understanding (Li et al., 2024; Zhang et al., 2024a,b), knowledge reasoning (Lyu et al., 2024; Xu et al., 2024), and recommender systems (Wu et al., 2024b; Zheng et al., 2023, 2024). However, some recent studies point out that LLMs are not ideal for NER directly (Han et al., 2023; Xie et al., 2023), and often need to decompose the task into multiple steps or continue to fine-tune on large-scale data (Wei et al., 2023; Xu et al., 2023; Zhou et al., 2024). These methods will undoubtedly consume a lot of resources.

Therefore, in this paper, we propose to leverage the collaboration of Small Language Models (SLMs) and LLMs to exploit their respective advantages: low resource consumption of SLMs and the extensive knowledge base of LLMs. We aim to address the non-target domain entity span problem inherent in SLMs while mitigating the high resource consumption of LLMs. Along these lines, we propose Double-Checker, a framework where the LLM functions as a checker. Instead of re-identifying entities, the LLM rechecks the candidates identified by the small model, ensuring more accurate and domain-specific entity recognition. Specifically, we first obtain the candidates predicted by the SLM

*Corresponding authors

[1]In Appendix A.1, we conduct a related experiment.

Findings of the Association for Computational Linguistics: EACL 2024, pages 3172–3181
November 12-16, 2024 ©2024 Association for Computational Linguistics

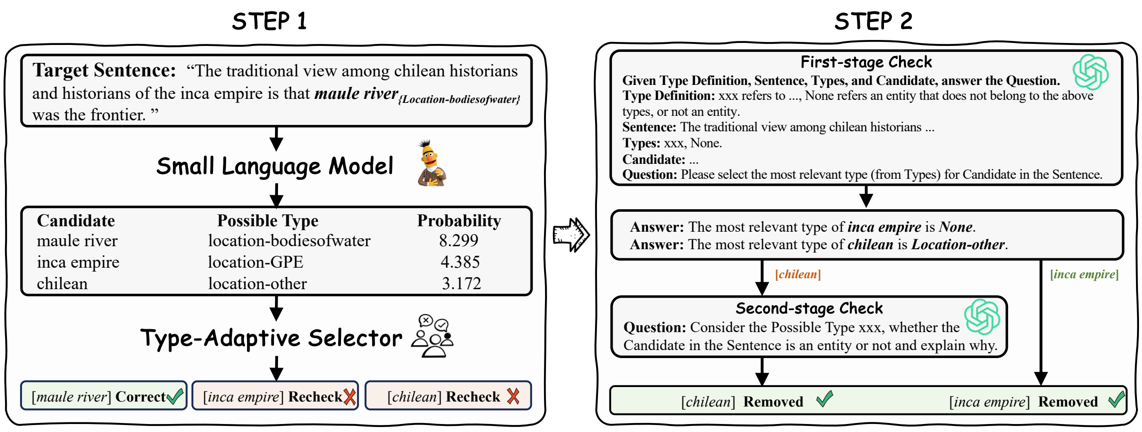

Figure 1: The overall framework of Double-Checker.

on the target domain sentences. To balance performance and resource consumption, we then utilize a type-adaptive selector to identify which candidates need to be rechecked. Finally, we use the LLM to conduct a two-stage check of the selected candidates, removing incorrectly identified spans to obtain the final results. We conduct extensive experiments on five few-shot NER datasets, achieving consistent performance improvements with the LLM on two state-of-the-art (SOTA) SLMs.

2 Methodology

In this section, we introduce Double-Checker, an efficient framework specifically designed for eliminating non-target domain candidates by rechecking the predictions made by small models. The framework consists of two main steps: firstly, we obtain the candidates predicted by the small model and select the ones to be rechecked; subsequently, a two-stage check utilizing LLM is conducted. An overview of the framework is shown in Figure 1.

2.1 Step 1: Select the Candidates

For each sentence x_i in the target domain, we first leverage the small model to obtain the structural output, denoted as $y_i = [s_i, t_i, p_i]$. Here, s_i represents a candidate span, t_i indicates the corresponding type, and p_i is the probability values.

Intuitively, outputs with higher predicted probability values are less likely to be incorrect. However, considering the high computational cost of using LLM, it is crucial to balance performance and cost by selecting an appropriate subset of data for the LLM to process. We assume that the probability distribution varies across different entity types and that prediction values for different types have varying levels of importance. Therefore, we develop a type-adaptive selector that prioritizes samples for LLM check based on the type-specific probability distributions, ensuring the most critical data is checked within the same data proportion. Specifically, we first construct a collection of probability values for each type:

$$Set(t_i) = \begin{cases} Set(t_i) \cup p_j, & \text{if } t_j = t_i \\ Set(t_i), & otherwise \end{cases}, \quad (1)$$

where $j \in \{1, ..., n\}$ (n is the number of candidates), t_j is the type of candidate j, and p_j is the predicted probability value. For each candidate, if its predicted type is t_i, we merge the probability value into $Set(t_i)$. Next, we set a quantile point α, which we assume to be 60%. If the probability value of a candidate exceeds the 60th percentile of the samples within its corresponding type set, it is considered less likely to be incorrect. Otherwise, it proceeds to the second step for further verification. By implementing this process, we effectively select the desired candidates.

2.2 Step 2: Two-stage Check

In this step, we utilize the rich external knowledge of the LLM to perform a two-stage check of the selected candidates.

Prompt Construction. Following Zhang et al. (2023), we transform the task into a QA format comprising five components: *Type Definition*, *Sentence*, *Types*, *Candidate*, and *Question*. Detailed specifications of this format are provided in Appendix A.2. It is important to highlight the introduction of *Type Definition* and the selection scope of *Types*, which we will cover later.

Models	Intra					Inter				
	1∼2 shot		5∼10 shot		Avg.	1∼2 shot		5∼10 shot		Avg.
	5 way	10 way	5 way	10 way		5 way	10 way	5 way	10 way	
Full Test set										
ProtoBERT* (Fritzler et al., 2019)	20.76±0.84	15.05±0.44	42.54±0.94	35.40±0.13	28.44	38.83±1.49	32.45±0.79	58.79±0.44	52.92±0.37	45.75
NNshot* (Yang and Katiyar, 2020)	25.78±0.91	18.27±0.41	36.18±0.79	27.38±0.53	26.90	47.24±1.00	38.87±0.21	55.64±0.63	49.57±2.73	47.83
StructShot* (Yang and Katiyar, 2020)	30.21±0.90	21.03±1.13	38.00±1.29	26.42±0.60	28.92	51.88±0.69	43.34±0.10	57.32±0.63	49.57±3.08	50.53
CONTaiNER* (Das et al., 2022)	41.51±0.07	36.62±0.04	57.83±0.01	51.04±0.24	46.75	50.92±0.29	47.02±0.24	63.35±0.07	60.14±0.16	55.36
DecomposedMeta* (Ma et al., 2022b)	49.48±0.85	42.84±0.46	62.92±0.57	57.31±0.25	53.14	64.75±0.35	58.65±0.43	71.49±0.47	68.11±0.05	65.75
HEProto* (Chen et al., 2023)	53.03±0.30	46.45±0.21	65.70±0.21	58.98±0.22	56.04	66.40±0.18	60.91±0.20	72.53±0.11	68.92±0.20	67.19
HEProto†	52.64	46.26	65.58	58.93	55.85	66.01	60.92	72.29	68.86	67.02
TadNER* (Li et al., 2023)	60.78±0.32	55.44±0.08	67.94±0.17	60.87±0.22	61.26	64.83±0.14	64.06±0.19	72.12±0.12	69.94±0.15	67.74
TadNER†	59.72	55.15	67.60	60.68	60.79	64.57	62.80	71.82	69.32	67.13
Sampled Test set										
GPT-3.5-turbo	53.69	47.07	54.59	49.36	51.18	46.26	42.68	51.81	49.09	47.46
HEProto†	52.94	46.55	65.35	58.90	55.94	65.42	60.89	72.10	69.28	66.92
TadNER†	60.13	55.02	67.62	60.75	60.88	64.38	62.92	71.67	69.54	67.12
Double-Checker$_{-HEProto}$	59.98	54.74	69.00	62.61	61.58	**68.58**	65.76	**73.49**	71.29	**69.78**
Double-Checker$_{-TadNER}$	**64.43**	**60.11**	**70.14**	**64.63**	**64.74**	66.09	**65.81**	73.03	**71.50**	69.11
Δ Double-Checker vs. HEProto	7.04 ↑	8.19 ↑	3.65 ↑	3.71 ↑	5.64 ↑	3.16 ↑	4.87 ↑	1.39 ↑	2.01 ↑	2.86 ↑
Δ Double-Checker vs. TadNER	4.13 ↑	5.09 ↑	2.52 ↑	3.88 ↑	3.86 ↑	1.71 ↑	2.89 ↑	1.36 ↑	1.96 ↑	1.99 ↑

Table 1: Comparison of performance on Few-NERD with the Micro-F1 metric(%). † indicates that the results are from our re-implementation with the same seed. $*$ denotes the results are obtained from Chen et al. (2023) and Li et al. (2023). The best results are in **bold**.

One crucial reason for introducing the concept of *Type Definition* is the variability in the range of entity types across different datasets, which poses a challenge for LLMs that are not inherently aware of this variability. By incorporating a type-specific description, we can enhance the LLM's focus and performance on a given dataset. To achieve this, we input the entire set of types from the datasets into the LLM simultaneously. This approach allows the LLM to consider the complete spectrum of entity types and generate tailored descriptions for each specific domain type. In Appendix A.3, we show the full description of the target domain types obtained from GPT-3.5-turbo.

We then define the scope of *Types*. Unlike re-ranking methods (Ma et al., 2023; Zhang et al., 2024c) that focus on calibrating false entity types, our approach aims to exclude non-target domain spans or non-entities. Consequently, in most scenarios, it suffices to include only the highest predicted type and "None" (indicating a non-target domain entity or non-entity) within type scope. In certain cases, we also incorporate the second most likely type predicted by the small model to enhance overall performance. In Section 3.4.2, we delve into the impact of varying the types scope on performance.

Two-stage Check Workflow. The right part of Figure 1 illustrates the workflow. For each selected candidate, we obtain the corresponding *Type Definition* based on its predicted type and input it into the LLM along with other necessary information from the prompt to obtain recheck results. If a candidate is determined as "None", it is removed and the process ends; otherwise, we proceed to the second stage of checking. The check in the second stage serves solely to determine whether the candidate is an entity. Based on the context in previous stage, we directly input the new *Question*. If the candidate is deemed as an entity, it will be included in the final result; otherwise, it will be excluded. Through the above process, we remove the false entity span and combine the unselected candidates to constitute the final result.

3 Experiments

3.1 Datasets and Experimental Setup

Few-NERD (Ding et al., 2021) is a standard few-shot NER dataset, which consists of 8 coarse-grained entity types and 66 fine-grained entity types. It is divided into Intra and Inter settings, and the entity types of the train set, dev set and test set are non-overlapping under each setting. In this case, the Intra setting is divided according to coarse-grained types, while the Inter is divided according to fine-grained types. Referring to Das et al. (2022), we additionally conduct the **Domain Transfer** experiment utilizing data from diverse domains. The training set was sourced from OntoNotes (General) (Ralph et al., 2013), while the test set comprised I2B2 (Medical) (Stubbs and Uzuner, 2015), CoNLL03 (News) (Sang and De Meulder, 2003), WNUT17 (Social Media) (Derczynski et al., 2017), and GUM (Zeldes, 2017) datasets.

Models	1 shot					5 shot				
	I2B2	CoNLL	WNUT	GUM	Avg.	I2B2	CoNLL	WNUT	GUM	Avg.
	Full Test set									
ProtoBERT* (Fritzler et al., 2019)	13.4±3.0	49.9±8.6	17.4±4.9	17.8±3.5	24.6	17.9±1.8	61.3±9.1	22.8±4.5	19.5±3.4	30.4
NNshot* (Yang and Katiyar, 2020)	15.3±1.6	61.2±10.4	22.7±7.4	10.5±2.9	27.4	22.0±1.5	74.1±2.3	27.3±5.4	15.9±1.8	34.8
StructShot* (Yang and Katiyar, 2020)	21.4±3.8	62.4±10.5	24.2±8.0	7.8±2.1	29.0	30.3±2.1	74.8±2.4	30.4±6.5	13.3±1.3	37.2
CONTaiNER* (Das et al., 2022)	21.5±1.7	61.2±10.7	27.5±1.9	18.5±4.9	32.2	36.7±2.1	75.8±2.7	32.5±3.8	25.2±2.7	42.6
DecomposedMeta* (Ma et al., 2022b)	15.5±3.0	61.2±9.2	27.7±5.3	20.3±4.2	31.2	19.8±2.6	75.2±5.8	29.8±3.9	33.5±2.4	39.6
TadNER* (Li et al., 2023)	39.3±3.8	70.4±10.6	32.8±4.8	24.2±4.1	41.7	45.2±2.3	80.5±3.6	34.5±4.6	35.1±2.2	48.8
TadNER†	38.51±4.89	69.21±9.37	33.94±5.25	23.23±3.96	41.22	44.79±2.53	79.93±3.57	34.46±3.98	35.85±1.76	48.75
	Sampled Test set									
TadNER†	38.68±6.53	71.33±9.02	35.01±5.13	23.23±3.67	42.06	46.07±3.14	80.44±2.86	35.60±3.15	35.86±1.77	49.49
Double-Checker$_{-TadNER}$	**40.50±7.94**	**72.35±7.58**	**39.85±3.97**	**25.36±4.33**	**44.52**	**48.91±3.13**	**80.76±2.43**	**40.39±2.71**	**38.03±1.59**	**52.02**
Δ Double-Checker vs. TadNER	1.82 ↑	1.02 ↑	4.84 ↑	2.13 ↑	2.46 ↑	2.84 ↑	0.32 ↑	4.79 ↑	2.17 ↑	2.53 ↑

Table 2: Comparison of performance on Domain Transfer with the Micro-F1 metric(%). † indicates that the results are from our re-implementation. $*$ denotes the results are obtained from Li et al. (2023). The best results are in **bold**.

Methods	Intra					Inter				
	1∼2 shot		5∼10 shot		Avg.	1∼2 shot		5∼10 shot		Avg.
	5 way	10 way	5 way	10 way		5 way	10 way	5 way	10 way	
Double-Checker$_{-TadNER}$	**64.43**	**60.11**	**70.14**	**64.63**	**64.74**	**66.09**	**65.81**	**73.03**	**71.50**	**69.11**
w/o Second-stage Check	63.67	59.41	69.60	63.97	64.16	65.01	65.05	72.07	70.91	68.26
w/o Type Definition	62.92	58.76	68.66	62.86	63.30	65.60	65.57	72.60	70.53	68.58
w/o Recheck	60.13	55.02	67.62	60.75	60.88	64.38	62.92	71.67	69.54	67.12

Table 3: Ablation study on Few-NERD with the Micro-F1 metric(%).

We choose the two SOTA methods (HEProto[2] (Chen et al., 2023) and TadNER (Li et al., 2023)) as our SLMs, and use GPT-3.5-turbo as the LLM for all experiments. Follow previous works (Ma et al., 2022b; Chen et al., 2023), we use the entity-level micro f1 score for evaluation, which requires both the predicted entity span and type to be correct.

3.2 Main Results

Considering the high cost of LLM, we sample the 10,000 sentences in each sub-setting on the full test set, and reproduce a portion of baselines with the same seed for a fair comparison. Table 1 and 2 show the main results of the comparison between our proposed Double-Checker and baselines. It is evident that Double-Checker achieves consistent improvements over both SLMs. Specifically from Table 1, there is a minimum increase of 1.39% and a maximum increase of 8.19% on HEProto, while it ranges from 1.36% to 5.09% on TadNER respectively. Furthermore, based on the average performance comparison, we observe that the improvement is more pronounced in the Intra setting due to a wider distribution gap between source and target domains where external knowledge provided by LLM effectively is better to bridges this. It is worth noting that GPT-3.5-turbo alone does not yield satisfactory results and even exhibits significant disparities compared to SOTA methods in most cases; however, when combined as part of Double-Checker, it not only consumes fewer resources but also achieves superior performance compared to both individual models.

[2]Due to the requirement for HEProto to leverage both coarse and fine granularity of labels, we opted not to utilize this model for our Domain Transfer experiments.

3.3 Ablation Study

We choose TadNER as SLM to conduct ablation experiments and introduce the following variants: 1) w/o Second-stage Check means that only the First-stage Check is retained. 2) w/o Type Definition removes the type definition in the prompt. 3) w/o Recheck denotes the origin results from SLM.

As shown in Table 3, we can observe that: 1) The removal of second-stage checks resulted in a decline in model performance, which validates the effectiveness of secondary reprocessing results. 2)When type definition are absent, Double-Checker drops more in Intra, indicating that enabling LLM to comprehend label ranges for more challenging tasks can better activate their internal knowledge.

3.4 Comparative Analysis

In this section, we conduct additional experiments to explore the following practical questions:

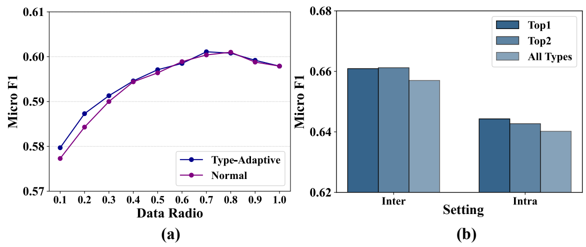

Figure 2: (a) Results of different selecting strategy. (b) Results of different types scope in prompt.

Q1: Why do we need type-adaptive selector?
Q2: How to adjust types scope in prompt?

3.4.1 Impact of Selector

We compare the performance of ours type-adaptive selector and normal selector (that is, selecting candidates based on all types) on Intra 10-way 1-shot. The Figure 2 (a) clearly demonstrates that our adaptive selector consistently outperforms in most cases, particularly when the proportion of selected data is low, thereby highlighting this phenomenon more prominently. Moreover, our method excels at selecting a greater number of non-target domain candidates with an equivalent data proportion. Additionally, it is worth noting that model performance does not always exhibit a linear relationship with the proportion of data; instead, it reaches a plateau and even declines. Consequently, considering both performance and resource consumption factors, the selector we have designed proves to be more suitable for realistic scenarios while achieving superior performance within limited resources.

3.4.2 Impact of Types Scope

The Figure 2 (b) illustrates a comparison of the impact of different types of scopes in prompt for 5-way 1-shot setting. It is evident that employing the full-type prompt yields the poorest results in both settings, whereas the other two options exhibit no significant differences. This can be attributed to a higher occurrence of errors in predicting non-target domain spans rather than type errors within the few-shot NER scenario. When providing a larger selection of types as input prompts to the large model, it inevitably introduces disturbances and shifts its objective from removing non-target domain spans to reclassifying spans, resulting in performance degradation. Therefore, for practical applications, it is advisable to limit the range of types provided as input prompts to minimize inference costs while potentially improving performance.

4 Conclusion

In this paper, we presented Double-Checker, a framework that effectively combines LLM and SLM for few-shot NER task. Specifically, we initially employed a type-adaptive selector to choose candidates predicted by the small model. Subsequently, the LLM is utilized to conduct a two-stage check process on these selected candidates, removing entity spans and non-entities that are not relevant to the target domain. Extensive experiments conducted using two different small models consistently demonstrated significant improvements, thereby showcasing the efficacy of our approach.

5 Limitations

Our approach aims to combine the complementary strengths of LLM and small models to enhance overall performance. Due to resource constraints, we are unable to run the LLM experiments on the entire test dataset (e.g., the Intra 10 way 5 shot setting includes over 300,000 sentences). Therefore, we sample 10,000 sentences for each setting. Another limitation is our lack of experimentation with additional LLMs, such as GPT-4 (Achiam et al., 2023) and LLaMA (Hugo et al., 2023). Exploring a broader range of LLMs could enable us to observe diverse experimental phenomena and facilitate more insightful analyses. We plan to address these limitations in a follow-up study.

Acknowledgments

This work was supported by the grants from National Natural Science Foundation of China (No.62222213, 62072423)

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM*

Transactions on Intelligent Systems and Technology, 15(3):1–45.

Wei Chen, Lili Zhao, Pengfei Luo, Tong Xu, Yi Zheng, and Enhong Chen. 2023. Heproto: A hierarchical enhancing protonet based on multi-task learning for few-shot named entity recognition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 296–305.

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)2*, pages 6338–6353.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, (Volume 1: Long Papers)*, pages 3198–3213.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *SAC*, pages 993–1000. ACM.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Touvron Hugo, Martin Louis, Stone Kevin, and others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xinhang Li, Jingbo Zhou, Wei Chen, Derong Xu, Tong Xu, and Enhong Chen. 2024. Visualization recommendation with prompt-based reprogramming of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13250–13262.

Yongqi Li, Yu Yu, and Tieyun Qian. 2023. Type-aware decomposed framework for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8911–8927.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2024. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *arXiv preprint arXiv:2401.17043*.

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.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022b. Decomposed meta-learning for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1584–1596.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

OpenAI. 2023. Introduce ChatGPT. *OpenAI blog*.

Ji Qi, Chuchun Zhang, Xiaozhi Wang, Kaisheng Zeng, Jifan Yu, Jinxin Liu, Lei Hou, Juanzi Li, and Xu Bin. 2023. Preserving knowledge invariance: Rethinking robustness evaluation of open information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 5876–5890.

Weischedel Ralph, Palmer Martha, Marcus Mitchell, Hovy Eduard, Pradhan Sameer, Ramshaw Lance, Xue Nianwen, Taylor Ann, Kaufman Jeff, Franchini Michelle, El-Bachouti Mohammed, Belvin Robert, and Houston Ann. 2013. Ontonotes release 5.0. *LDC2013T19, Philadelphia, Penn.: Linguistic Data Consortium*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 142–147.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of biomedical informatics*, 58:S20–S29.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 5012–5024.

Xiangfeng Wang, Zaiyi Chen, Zheyong Xie, Tong Xu, Yongyi He, and Enhong Chen. 2024. In-context former: Lightning-fast compressing context for large language model. *arXiv preprint arXiv:2406.13618*.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with ChatGPT. *arXiv preprint arXiv:2302.10205*.

Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2024a. Exploring large language model for graph data understanding in online job recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9178–9186.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024b. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot NER with ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Derong Xu, Ziheng Zhang, Zhihong Zhu, Zhenxi Lin, Qidong Liu, Xian Wu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024. Editing factual knowledge and explanatory ability of medical large language models. *arXiv preprint arXiv:2402.18099*.

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*, pages 6365–6375.

Amir Zeldes. 2017. The gum corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Chao Zhang, Shiwei Wu, Haoxin Zhang, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024a. Notellm: A retrievable large language model for note recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 170–179.

Chao Zhang, Haoxin Zhang, Shiwei Wu, Di Wu, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024b. Notellm-2: Multimodal large representation models for recommendation. *arXiv preprint arXiv:2405.16789*.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812.

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. 2024c. Linkner: Linking local named entity recognition models to large language models using uncertainty. In *Proceedings of the ACM on Web Conference 2024*, pages 4047–4058.

Lili Zhao, Qi Liu, Linan Yue, Wei Chen, Liyi Chen, Ruijun Sun, and Chao Song. 2024. Comi: Correct and mitigate shortcut learning behavior in deep neural networks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 218–228.

Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. 2024. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3207–3216.

Zhi Zheng, Zhaopeng Qiu, Xiao Hu, Likang Wu, Hengshu Zhu, and Hui Xiong. 2023. Generative job recommendations with large language model. *arXiv preprint arXiv:2307.02157*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

A Appendix

A.1 Interference from the Source Domain

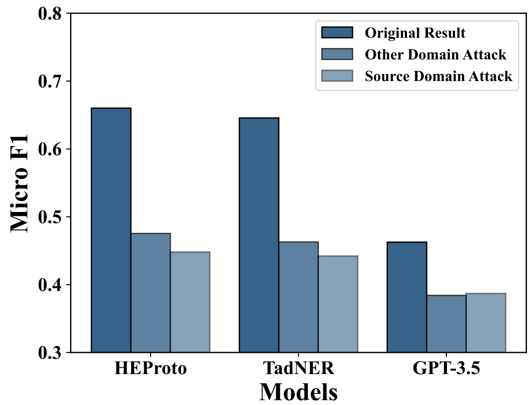

Figure 3: Results of different domain "entity attacks" on SOTA methods and GPT-3.5 on Few-NERD dataset. "Other Domain" denotes the dev set, and "Source Domain" refers to the train set.

The domain-agnostic detector is affected by the source domain. In order to demonstrate this phenomenon, we re-constructed the target domain (test set) of Few-NERD (Ding et al., 2021) by "entity attacks". Specifically, we first collect entity sets from the source domain (train set) and other domain (dev set). Then, we randomly select a non-entity position in the target domain sentence and insert entities from the two domains separately, thus constructing two interference datasets for entity attacks from different domains. As shown in the Figure 3, we can observe that all the models have a huge drop in performance on two interference datasets. Notice that the BERT-based models have proportionally more performance degradation compared to the GPT-3.5 (OpenAI, 2023) and are subject to more interference from source domain attacks. We attribute this to the fact that the BERT-based model needs to absorb the knowledge of the source domain during the training process, and the span detector fine-tuned on the target domain cannot completely get rid of the influence of the source domain knowledge, leading to easier detection of entity spans in the non-target domain. Large Language Models, on the other hand, possess rich internal knowledge and are naturally more resistant to interference (Achiam et al., 2023; Qi et al., 2023; Chang et al., 2024; Zhao et al., 2024; Wu et al., 2024a).

A.2 Prompt Example

We select a candidate from target sentence and construct the corresponding prompt, the details of which are shown in the Table 4.

A.3 Type Definition Example

We use GPT-3.5-turbo to generate the full type definitions in the test set, with some examples presented in the Table 5 and 6.

<*First-stage Check Prompt*>

Given the Type Definition, Sentence, Types, and Candidate, answer the Question.

Type Definition: location-GPE includes names of countries, cities, states, provinces, and other regions that have a political or geographical significance. None refers an entity that does not belong to the above types, or is not an entity.

Sentence: he was born into a christian family in the predominantly muslim north.

Types: location-GPE, None

Candidate: muslim north

Question: Please refer to Type Definition and select the most relevant type (from Types) for Candidate in the Sentence. Answer in the format of json like: {"answer": " "}

<*Second-stage Check Prompt*>

Question: Consider the Possible Type {first-stage answer}, whether the Candidate in the Sentence is an entity or not. Answer in the format of json like: {"answer": " "}

Table 4: An example of the prompt of our two-stage check.

Type Defnition
location-GPE includes names of countries, cities, states, provinces, and other regions that have a political or geographical significance.
location-other is a catch-all category within the location entity type that includes geographical locations which do not fit into the more specific subcategories listed.
location-mountain refers to geographical entities that are elevated landforms characterized by steep slopes, rocky terrain, and often having peaks or summits.
location-bodiesofwater refers to geographical entities that are large bodies of water, such as oceans, seas, rivers, lakes, and other water reservoirs.
location-island refers to geographical entities that are landmasses surrounded by water on all sides.
location-park refers to designated areas of land that are preserved or managed for recreational, conservation, or aesthetic purposes.
location-road/railway/highway/transit refers to infrastructure designed for transportation, including roads, railways, highways, and transit systems.
organization-education refers to institutions or entities primarily focused on providing education and academic instruction.
organization-government/governmentagency refers to entities that are part of or associated with governmental bodies and agencies.
organization-company refers to entities that are businesses or commercial enterprises. This category includes names of companies, corporations, firms, and other types of business organizations.
organization-politicalparty refers to entities that are organized groups of people with similar political aims and opinions.
organization-other is a category within the organization entity type that includes organized groups or entities which do not fit into the more specific subcategories listed.
organization-media/newspaper refers to entities involved in the production and dissemination of news and information to the public through various media channels.
organization-religion refers to entities associated with religious beliefs, practices, and institutions.
organization-showorganization refers to entities involved in the production, promotion, or organization of entertainment events and performances.
organization-sportsleague refers to entities that are structured groups or associations governing a particular sport or a group of sports.
organization-sportsteam refers to entities that are teams participating in competitive sports, usually within the structure of a sports league or association.

Table 5: Definition of types on the target domain of Few-NERD Intra.

Type Defnition
other-medical refers to entities, concepts, or items related to the field of medicine that do not fit into more specific categories.
person-athlete refers to individuals who engage in physical sports or other forms of competitive physical activities.
event-sportsevent refers to organized competitive events or activities in which athletes or teams participate in sports.
art-music refers to entities and works associated with the creation, performance, and recording of music.
other-livingthing refers to entities that are living organisms but do not fit into more specific categories like humans, specific animals, or plants.
building-hospital refers to structures specifically designed and equipped for the delivery of healthcare services.
building-theater refers to structures specifically designed for the performance of live entertainment, such as plays, musicals, dance performances, concerts, and other stage productions.
other-educationaldegree refers to academic qualifications or titles that do not belong to more specific categories within the educational domain.
person-actor refers to individuals who professionally perform roles in plays, films, television shows, or other forms of entertainment media.
product-car refers to automobiles or vehicles designed for transportation purposes.
product-weapon refers to devices or instruments designed or used for inflicting harm, damage, or destruction.
art-writtenart refers to artistic works that are expressed through the written word.
event-election refers to the process of selecting individuals for specific roles or positions through a structured voting system.
None refers an entity that does not belong to the above types, or is not an entity.

Table 6: Definition of types on the target domain of Few-NERD Inter, some of which are described in Table 5.