

Scaling Performance and Low-Resource Annotation with Many-Shot In-Context Learning for Named Entity Recognition

Qi Zhang¹ Fangping Lan¹

Cornelia Caragea² Longin Jan Latecki¹ Eduard Dragut¹

¹Temple University ²University of Illinois Chicago

{qi.zhang, latecki, edragut}@temple.edu, cornelia@uic.edu

Abstract

In-context learning (ICL) with large language models (LLMs) has emerged as a powerful alternative to fine-tuning for Named Entity Recognition (NER), achieving strong performance with minimal annotation and no additional training. However, prior work has shown that despite their adaptability, LLMs still lag behind fully supervised models such as fine-tuned BERT in structured tasks like NER. While existing studies on ICL for NER have mainly explored few-shot settings, the potential of scaling to hundreds of demonstrations has not been thoroughly investigated. To address this gap, we conduct a comprehensive investigation of many-shot ICL for NER and further explore its effectiveness in annotating and refining data for low-resource NER tasks. Specifically, we evaluate various LLMs across multiple domains using hundreds of ICL examples and then assess the feasibility of using many-shot ICL as a data annotation framework. Our experiments demonstrate that: (1) scaling to hundreds of in-context examples enables LLMs to match or even surpass the performance of fully supervised BERT models; and (2) using about one hundred human-labeled examples as demonstrations, many-shot in-context annotation can generate high-quality labeled data, leading to approximately 10% absolute F1 improvement over existing state-of-the-art approaches when used to fine-tune BERT on low-resource NER¹.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in information extraction that aims to identify and classify entities such as persons, organizations, locations, and domain-specific concepts within text (Jurafsky, 2000; Pan et al., 2025b). It serves various downstream applications, including knowledge base and knowledge graph construction (Zhang

et al., 2024a; Wei et al., 2024), question answering (Choudhary et al., 2023), and domain-specific document understanding (Singh et al., 2023; Nandi and Agrawal, 2024). State-of-the-art (SOTA) NER are small models such as BERT fine-tuned on in-domain data (Hu et al., 2024a; Zhang et al., 2024b, 2022). They require large amount of human-labeled examples (Devlin et al., 2019; Ding et al., 2021; Sun et al., 2025). A typical workaround is to generate large-scale labeled instances using rule-based or heuristic functions (Pryzant et al., 2022; Safranchik et al., 2020; Shang et al., 2018), followed by fine-tuning with a smaller set of human-labeled samples. In this work, we explore the use of large language models (LLMs) for in-context learning (ICL) to perform NER without updating model parameters (Wang et al., 2025; Ashok and Lipton, 2023; Pan et al., 2025a). Our key insight is that with a moderate number of human-annotated examples (e.g., 100), LLMs can act as in-context annotation functions, generating thousands of high-quality, labeled samples for training smaller, more efficient models.

While LLMs in few-shot settings still underperform compared to fully supervised small models (Xie et al., 2024; Heng et al., 2024) and are slower at inference (Zaratiana et al., 2024; Bogdanov et al., 2024), they excel at generalizing from a few examples, making them ideal for bootstrapping annotations in low-resource domains. Rather than relying on LLMs as full-time NER systems, we leverage them to generate training data for more efficient small models. This approach is made feasible by the increased context length in modern LLMs, which supports many-shot ICL—scaling the number of demonstrations from a few to several hundred (Fu et al., 2024; Gao et al., 2024). Prior work shows that ICL performance improves with more demonstrations (Bertsch et al., 2025; Song et al., 2025; Agarwal et al., 2024), yet many-shot ICL paradigm remains underexplored in NER.

¹Code and data are available at [Many-Shot-ICL-NER](#).

We propose an In-Context Annotation (ICA) framework that leverages many-shot ICL to automatically label domain-specific NER datasets. In low-resource settings, ICA uses 100 human-labeled examples to annotate 2k unlabeled instances. To enhance annotation quality, we apply refinement strategies such as self-consistency, self-correction, and error-aware self-refinement (ERA). The annotated data is then used to fine-tune a BERT-based model, which outperforms baselines by up to 10 F1 points—achieving competitive results with substantially less human effort.

We conduct a comprehensive study of many-shot ICL for NER across four domains, evaluating both open-source and closed-source LLMs. Our experiments show that using around 100 in-context examples allows LLMs to match or even surpass the performance of supervised models trained on thousands of human-annotated instances, demonstrating the data efficiency of the ICA framework. Finally, we analyze the scalability of ICA, the impact of seed set size, and the potential of using smaller or open-source models for annotation. These findings offer practical insights into the strengths and limitations of LLMs as efficient NER annotators, guiding future research and deployment. To sum up, our contributions are four-fold:

- **Empirical Study.** We present the first systematic study of *many-shot* in-context learning (ICL) for NER, showing consistent, monotonic gains as the number of demonstrations increases across four datasets and seven LLMs (§2).
- **Framework.** We propose a *many-shot In-Context Annotation (ICA)* framework that relies on a modest number of human-labeled samples. ICA includes several refinement strategies to enhance annotation quality (§3.1).
- **Experiments.** Through experiments on five low-resource NER datasets, we show that our approach outperforms SOTA baselines by a substantial margin, achieving an average absolute F1 improvement of approximately 10 points (§3.3).
- **Analyses.** We conduct extensive analyses on the scalability of ICA, the efficiency of seed annotation, and the feasibility of using smaller, open-source LLMs for annotation tasks (§4).

2 Many-Shot In-Context Learning NER

In this section, we first describe the task formulation of ICL for NER in §2.1, and then present our experimental setup and results in §2.2.

Task: Identify and tag entities in the input text by wrapping them with XML-like tag `<entity type="[TYPE]">...</entity>` tags. Replace [TYPE] with the correct entity category. Make sure that you only wrap text that corresponds to an entity and leave the rest of the text unchanged. Tag all valid entities, even if they repeat.

Entity Types: Person, Location, Organization, Miscellaneous

Examples: {examples}

Test Data:
 Input: Canada 's were the worst performing bonds .
 Output: `<entity type="location">Canada</entity>` 's were the worst performing bonds .

Figure 1: XML-style tagging prompt. {examples} denotes the in-context demonstrations, each following the same input-output format.

2.1 In-Context Learning for NER

We formulate the NER task as a conditional generation problem. Given an input sentence, the goal is to generate its annotated version in a structured format where entity spans and type are explicitly marked. This formulation enables the use of LLMs via ICL, allowing the model to learn the labeling behavior from a set of demonstration examples and apply it to test instances. Let $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ denote the training and test datasets, respectively. We construct the ICL demonstration set by sampling k labeled examples from the training set:

$$\mathcal{D}_{\text{ICL}} = \{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\} \subset \mathcal{D}_{\text{train}} \quad (1)$$

where x_i is an input sentence and y_i is its output in the target format. Given a test input $x_{\text{test}} \in \mathcal{D}_{\text{test}}$, we construct the prompt as:

$$\text{Prompt} = \text{I} + \sum_{i=1}^k \text{Format}(x_i, y_i) + \text{Format}(x_{\text{test}}) \quad (2)$$

where I denotes the task instruction, and $\text{Format}(x, y)$ renders a labeled example in a consistent input-output style. The model then generates:

$$\hat{y} = \text{LLM}(\text{Prompt}) \quad (3)$$

which aims to recover the entity-annotated version of x_{test} . Formally, we model the generation of the output as a conditional probability over the instruction, the demonstration set, and the test input:

$$\hat{y}_{\text{test}} = \arg \max_y P(y \mid \text{I}, \mathcal{D}_{\text{ICL}}, x_{\text{test}}) \quad (4)$$

XML-style Output Format. To represent both entity spans and their types in generation, we adopt an XML-style annotation format in which each entity mention is wrapped with explicit tags that include the entity type as an attribute. Figure 1 shows

the prompt used in our ICL for NER. This XML-style generation format, which explicitly marks both entity boundaries and types, and preserves surrounding non-entity tokens, has been shown to improve the boundary sensitivity of LLMs and reduce span-level annotation errors (Ding et al., 2024; Zhang et al., 2024a; Hu et al., 2024b). Moreover, the structured output can be easily converted into the traditional Beginning-Inside-Outside (BIO) format, allowing direct comparison with results from previous NER systems. We adopt this XML-style format as the default prompting template throughout this work.

2.2 Experiments

Datasets. To ensure diversity and alignment with real-world application scenarios, we evaluate many-shot ICL on four NER datasets from different domains: MIT-Restaurant, MIT-Movie (Liu et al., 2013), WNUT 2017 (Derczynski et al., 2017), and CoNLL2003 (Tjong Kim Sang and De Meulder, 2003). CoNLL2003 is a widely used benchmark for standard NER tasks in general domains. MIT-Restaurant and MIT-Movie are domain-specific NER datasets with 8 and 12 fine-grained entity types, respectively, posing a greater challenge due to their specialized vocabulary and entity distribution. WNUT 2017 focuses on NER in noisy and informal social media texts, further testing a model’s robustness under real-world conditions. More dataset details are provided in Appendix A.1.

Models and Settings. We evaluate a diverse set of LLMs, including open-source models such as Qwen-2.5 (7B, 32B, and 72B) (Yang et al., 2025a) and LLaMA-3.1 (8B and 72B) (Grattafiori et al., 2024), as well as frontier models like GPT-4o (OpenAI, 2024) and DeepSeekV3 (DeepSeek, 2024). All models support a context length of at least 32K tokens, which enables them to accommodate hundreds of in-context examples. For each dataset, we gradually increase the number of demonstrations up to 500 to study the scaling behavior of many-shot ICL. To select in-context examples, we consider two strategies: (1) **random sampling**: randomly sampling k examples from the training set as fixed demonstrations for the entire test set; and (2) **retrieval-based**: selecting the top- k training instances most similar to each test sample based on lexical similarity using BM25. For the supervised baseline, we fine-tune a BERT-based model on the full training set of each dataset, referred to as

BERT-FT. We additionally include a target-only variant, **BERT-FT (target-only)**, which is fine-tuned solely on the limited human-annotated examples available in each low-resource target domain (100–200 samples), without any access to source-domain data. This variant serves as a lower-bound reference that isolates the contribution of our ICA-based data generation from any source-domain transfer. To assess the impact of LLM choice on downstream annotation quality, we compare annotations generated by GPT-4o, DeepSeekV3, and three variants of Qwen-2.5 (7B, 32B, 72B) in our analysis (§4). All reported scores are entity-level micro-F1 with exact span matching, computed using the `seqeval` package under default settings. Appendix A.2 describes the experimental settings.

Results. Due to page constraints, we present a subset of our results in Figure 2. Comprehensive experimental results and detailed score tables are in Appendix A.4. Across all domains, we observe a consistent improvement in F1-score as the number of in-context demonstrations increases. This trend is especially evident for the retrieval-based ICL setting, where performance improves rapidly and plateaus around 100 examples. An exception is observed on CoNLL2003, where performance saturates even earlier (around 50 examples). In contrast, randomly sampled demonstrations yield a more gradual and steady improvement as the number of examples grows. Notably, for the same model, the performance gap between retrieval-based and random sampling narrows as more examples are included, with the two approaches showing comparable results when scaled to 500-shot.

While retrieval-based ICL consistently outperforms random sampling in most cases (except WNUT2017 with 500 samples), it is important to highlight the cost difference between the two settings. Random sampling selects a fixed set of k demonstrations once and reuses them for all test examples. In contrast, retrieval-based ICL involves dynamically selecting top- k nearest neighbors for each test instance from the training pool, resulting in significantly higher annotated data requirement. Our results suggest that although random sampling lags behind in performance initially, increasing the number of random samples can yield highly competitive results, approaching those of the more expensive retrieval-based approach.

Comparing the many-shot ICL performance against BERT-FT, we found that in nearly all cases,

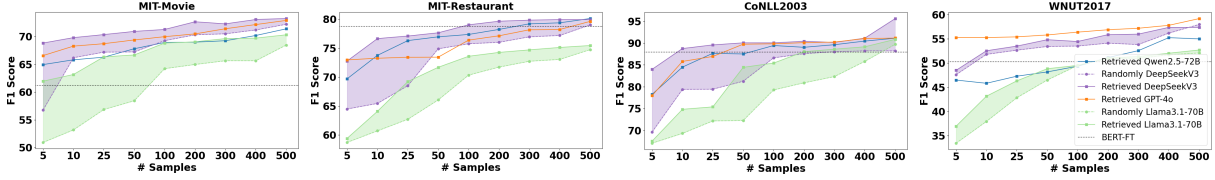


Figure 2: Performance trends on the MIT-Movie, MIT-Restaurant, CoNLL2003 and WNUT2017 datasets as the number of samples increases.

except for LLaMA3.1-70B on MIT-Restaurant, LLMs achieve performance comparable to or even exceeding that of BERT-FT on the full training set. This highlights the strong data efficiency of many-shot ICL, demonstrating that LLMs can match supervised performance using only hundreds of labeled demonstrations without additional parameter updates. This finding is especially important for random sampling, which achieves competitive performance while requiring much less annotation effort than both BERT-FT and retrieval-based ICL. For more results and complete per-model evaluation tables, refer to Appendix A.4.

3 Many-Shot In-Context Annotation for Low-Resource NER

The results in Section 2 show that many-shot in-context learning enables LLMs to achieve strong performance in NER with only around hundreds of labeled examples. However, despite its effectiveness, many-shot ICL requires significant inference cost—each new prediction involves long prompts and repeated calls to large models. This motivates a practical alternative: instead of relying on LLMs for expensive real-time inference, we consider employing them as annotators to generate high-quality labeled data that can be used to train smaller, more efficient models like BERT. Motivated by this, we hypothesize that if a small number of labeled examples can guide high-quality ICL predictions, then the same setup can be used to automatically annotate unlabeled corpora. We consider a realistic setting in which only a modest set of human-annotated examples (e.g., 100) is available. We aim to use these examples as fixed in-context demonstrations to prompt an LLM and annotate thousands of unlabeled sentences. We refer to this process as **Many-shot In-Context Annotation (ICA)**, where a fixed set of human-labeled examples is used as in-context demonstrations to prompt an LLM for annotating large-scale unlabeled data.

In this section, we first introduce our proposed ICA framework in §3.1, which generates struc-

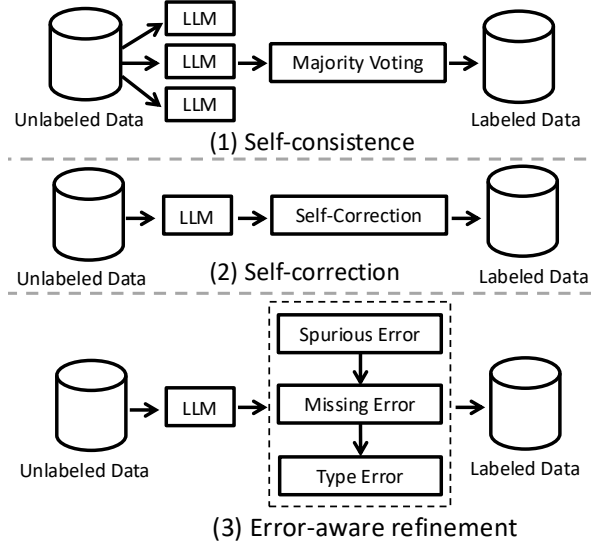


Figure 3: Three refinement methods used in our proposed ICA framework.

tured annotations for each sentence using the same fixed prompt, enabling the training of a downstream BERT-based NER model. To further enhance annotation quality, we propose three strategies: **self-consistency**, **self-correction**, and a **error-aware refinement (ERA)** method aimed at correcting common NER-specific annotation errors. We describe the experimental setup in §3.2, followed by results across five low-resource domains in §3.3.

3.1 ICA Framework

We describe here our ICA framework, which builds upon the ICL formulation introduced in § 2.1. Given a small set of human-annotated examples, we use them as fix demonstrations to prompt an LLM and annotate a much larger unlabeled corpus. The generated annotations follow the same XML-style output format as described in § 2.1, and are converted to BIO format to train a downstream BERT-based NER model. To further improve the quality of annotated data, we introduce a refinement step based on LLM confidence. Specifically, we calculate the average log probability over the entity tokens enclosed by XML tags, excluding all non-entity tokens. Formally, for a generated sentence \hat{y} annotated by the LLM by Eq. 3 to in-

clude entity spans $\{s_1, s_2, \dots, s_n\}$, where each s_i consists of a token sequence representing an entire XML-style entity annotation (e.g., `<entity type="type">mention</entity>`), we define the confidence score as:

$$\text{Confidence}(\hat{y}) = \frac{1}{\sum_{i=1}^n |s_i|} \sum_{i=1}^n \sum_{t \in s_i} \log p(t) \quad (5)$$

where $|s_i|$ denotes the number of tokens in the i -th entity span, and $p(t)$ is the generation probability of token t . Consider the example sentence in Figure 1, the confidence score is computed using the log probabilities of the entire entity span `<entity type="location">Canada</entity>`. This provides a more focused estimate of annotation confidence.

We begin with a standard ICA setup, where each unlabeled sentence is annotated via a single LLM call using a fixed prompt with human-labeled demonstrations. To improve annotation quality, we further introduce three refinement strategies: self-consistency, self-correction, and error-aware refinement (EAR). They are applied to the bottom 50% of low-confidence samples obtained by Eq. 5. As shown in Figure 3, each strategy builds on the initial ICA output in different ways: (1) Self-consistency performs two additional LLM predictions for each low-confidence sentence, using varied sampling and demonstration orders, and aggregates the three outputs via majority voting over both entity spans and types. (2) Self-correction introduces a second prompting step where the LLM is explicitly asked to review and revise its own annotations. (3) EAR applies three dedicated correction prompts in sequence to handle common NER-specific errors, including spurious entities, missing entities, and incorrect entity types. Due to space constraints, detailed implementation and prompt designs are in Appendix B.1.

3.2 Experimental Settings

Dataset. We conduct experiments on the CrossNER (Liu et al., 2021) benchmark, which covers five diverse domains: *AI*, *Literature*, *Music*, *Politics*, and *Science*. We choose CrossNER as our testbed because it presents a realistic and challenging low-resource scenario: each domain contains a large number of entity types but only 100 or 200 labeled examples. Moreover, the small size of the labeled data reduces the risk of data contamination from LLM pretraining corpora (Deng et al., 2024),

making it a cleaner and more trustworthy benchmark for evaluating low-resource NER methods. For each domain, we randomly sample 100 human-annotated sentences from the provided training set to serve as our in-context demonstration set. These examples are used to prompt the LLM for annotation. Detailed statistics of each domain are provided in Appendix B.2.

Unlabeled Data Collection. Inspired by prior retrieval-augmented data augmentation methods (Cai et al., 2023), we use BM25 to retrieve the top-10 most similar unlabeled sentences for each of the 100 seed examples. This yields a set of 1k sentences that are semantically close to the labeled data and likely to contain relevant entity contexts. To further improve coverage and diversity, we additionally sample 1k sentences uniformly at random from the remaining unlabeled corpus (excluding the retrieved sentences). The unlabeled corpus in each domain is sourced from the CrossNER benchmark (Liu et al., 2021), which provides domain-specific Wikipedia sentences. This setup is consistent with prior data augmentation work that also leverages Wikipedia-derived corpora (Cai et al., 2023). The final annotation set for each domain therefore contains 2k unlabeled sentences, combining high-relevance and high-diversity examples.

Evaluation. We use the DeepSeekV3 to perform annotation over the 2k unlabeled sentences collected as described above. The generated annotations are converted into the standard BIO format and used to train a BERT-based NER model. To evaluate the effectiveness of the proposed ICA, we compare our approach with several representative baselines from two categories. The first group includes traditional data augmentation methods such as DAGA (Ding et al., 2020), NERDA (Dai and Adel, 2020), and GPDA (Cai et al., 2023) and LLM-based data augmentation methods ProgGen (Heng et al., 2024). The second group includes recent state-of-the-art methods for low-resource or cross-domain NER, such as LST-NER (Zheng et al., 2022), DoSEA (Tang et al., 2022), Hu et al. (2022), DTrans-SMix (Zhang et al., 2022), DH-GAT (Xu and Cai, 2023), Three-KPNs (Hu et al., 2024a), DTrans-MPrompt (Zhang et al., 2024b), PromptNER (Ashok and Lipton, 2023), IF-WRANER (Nandi and Agrawal, 2024), GliNER (Zaratiana et al., 2024), and B2NER (Yang et al., 2025b). We also include the performance of using LLM to do in-context inference with zero-shot (ICL-ZS) and

100-shot (ICL-MS) for comparison. More details about the baselines implement are provided in Appendix A.3.

3.3 Results

Table 1 presents the performance of our proposed ICA-based annotation strategies, compared with multiple strong baselines, including LLM-based inference, traditional data augmentation methods, and recent SOTA low-resource NER approaches. We report F1-scores averaged over five runs across five domains from the CrossNER benchmark.

In summary, our results provide strong evidence that **LLM-assisted In-Context Annotation** substantially outperforms both traditional data augmentation techniques and SOTA low-resource NER methods. Importantly, our framework relies solely on 100 human-annotated examples per domain, whereas prior methods typically require additional development sets and access to source-domain annotated corpora. *These findings highlight the power of pre-trained LLMs and their in-context learning capabilities for efficient and scalable dataset construction in low-resource settings.* They suggest that combining a small amount of acceptable human supervision with powerful LLMs via in-context learning can serve as a practical and scalable paradigm for building NER datasets in resource-constrained domains.

Compared to direct in-context inference using LLMs (ICL-MS), our ICA-based approach offers both higher performance and greater practical utility. While ICL-MS achieves strong results (79.44 average F1), our best ICA variant (ICA with EAR) improves this by over 5 F1-scores, indicating that converting LLM predictions into training data allows for more effective downstream models training. Additionally, ICA enables the use of compact models like BERT for inference, which are significantly more efficient to deploy than prompting large LLMs with lengthy demonstrations (Bertsch et al., 2025). This makes ICA more suitable for real-world NER applications, particularly in resource-constrained settings.

Finally, we find that applying targeted refinement strategies further improves the quality of ICA. Among them, the proposed Error-Aware Refinement (EAR) consistently outperforms other variants across all domains, yielding an average gain of 1.79 F1 points over standard ICA. This demonstrates that explicitly addressing common LLM annotation errors, such as missing entities, spu-

rious entities, and type mismatches, can lead to more accurate and reliable supervision for downstream model training. We further include a case study in Appendix B.3 to qualitatively illustrate the effectiveness of these refinement strategies. We also provide a detailed cost analysis and further discussion on deployment scalability of our ICA approach in Appendix B.5

Effectiveness of Refinement at Inference Time.

A natural question is whether our refinement strategies also help when applied directly at inference rather than as a data-annotation step. We apply Self-Correction and EAR to ICL-MS outputs (see Table 1). ICL-MS w/ EAR improves over ICL-MS by +2.41 average F1 (79.44 \rightarrow 81.85), confirming that our refinement strategies are effective beyond the data-annotation setting. We note that prior work has observed limitations of generic intrinsic self-correction in zero-shot reasoning tasks (Huang et al., 2023; Hao et al., 2025). Our findings, in line with domain-specific studies such as ProgGen (Heng et al., 2024), suggest that these limitations can be mitigated by (i) replacing open-ended self-verification with task-specific few-shot demonstrations of error patterns paired with corrections, and (ii) decomposing refinement into focused sub-tasks (spurious, missing, type) rather than asking the model to fix everything at once. Nevertheless, the ICA-based approach still outperforms ICL-MS w/ EAR by 2.95 F1, indicating that converting LLM predictions into downstream training data yields additional gains beyond what refinement alone provides, while also enabling efficient BERT-based deployment.

4 Additional Studies and Discussion

In this section, we report a series of experiments to further understand the behavior and requirements of LLMs when used for in-context annotation in low-resource NER settings. We aim to answer the following three research questions: **RQ1:** How much LLM-annotated data is needed to train an effective small NER model? **RQ2:** How much human-annotated data is required for LLMs to generate high-quality annotations? **RQ3:** How important is the choice of LLM? Can smaller or open-source models perform competitively? We conduct all studies on the *AI* and *Literature* domains and using ICA with EAR to annotate the data. In the following content of this section, “Previous SOTA” refers to DTrans-MPrompt (Zhang et al., 2024b),

Method	AI	Literature	Music	Politics	Science	Average
LLM In-Context Inference						
ICL-ZS	59.68	61.49	49.03	72.25	68.44	62.18
ICL-MS	74.82	77.83	83.72	83.05	77.79	79.44
ICL-MS w/ Self-Correction	76.15	78.92	84.65	83.94	78.85	80.50
ICL-MS w/ EAR	77.95	80.35	85.68	85.15	80.12	81.85
State-of-the-art Approaches						
LST-NER	63.28	70.76	76.83	73.25	70.07	70.84
DoSEA	66.03	68.59	73.10	71.69	75.52	70.99
DTrans-SMix	68.93	69.22	76.10	76.70	72.35	72.66
DH-GAT	69.30	72.51	78.77	77.06	74.21	74.37
B2NER	59.00	63.70	68.60	67.80	72.00	66.22
GLiNER [†]	70.55	75.13	83.27	81.11	75.73	77.16
Three-KPNs	70.05	70.39	77.57	77.82	73.63	73.89
PromptNER	68.83	74.44	84.26	78.61	72.59	74.95
DTrans-MPrompt	70.13	73.51	79.54	80.54	73.06	75.36
IF-WRANER	68.81	75.52	85.43	75.31	79.80	76.97
Data Augmentation Approaches						
DAGA	66.77	71.15	78.48	73.30	73.07	72.55
NERDA	70.20	71.28	79.56	75.30	74.37	74.14
GPDA	70.05	72.34	80.16	75.95	75.55	74.81
ProgGen	71.56	73.31	85.14	84.19	78.25	78.49
LLM-assisted In-Context Annotation (Ours)						
ICA	76.97 \pm 0.42	80.95 \pm 0.38	87.95 \pm 0.29	87.50 \pm 0.35	81.71 \pm 0.41	83.01 \pm 0.37
ICA w/ Self-Consistency	77.40 \pm 0.36	82.03 \pm 0.31	88.51 \pm 0.25	88.22 \pm 0.28	82.35 \pm 0.33	83.70 \pm 0.31
ICA w/ Self-Correction	77.79 \pm 0.28	81.89 \pm 0.25	89.53 \pm 0.22	87.69 \pm 0.26	82.02 \pm 0.29	83.78 \pm 0.28
ICA w/ EAR	79.88\pm0.21	82.96\pm0.18	89.30\pm0.15	88.69\pm0.19	83.17\pm0.23	84.80\pm0.18

Table 1: F1 scores on CrossNER. Results for our ICA methods are averaged over five runs with different random seeds; subscripts denote standard deviation. [†]GLiNER is fine-tuned with gliner_large-v2.5 on 100 human-annotated examples per domain using the official training script.

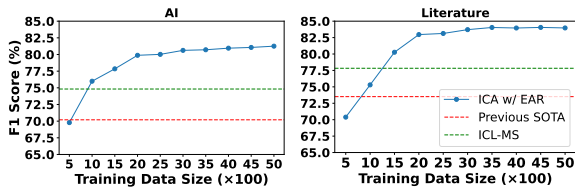


Figure 4: Performance trends on the AI and Literature domains as the annotated training data increases.

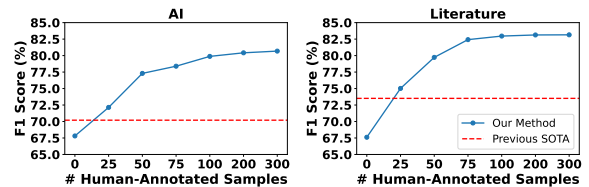


Figure 5: Performance trends on AI and Literature as the number of human-annotated samples increases.

the strongest cross-domain baseline from Table 1.

4.1 RQ1: How Much LLM-Annotated Data is Needed?

To investigate how much LLM-annotated data is needed to train an effective NER model, we conduct an annotation scale-up experiment. Specifically, we use ICA with EAR to label between 0.5k and 5k unlabeled sentences in increments of 0.5k. The resulting data is then used to train a BERT-based NER model, and we evaluate its performance on the test set of the corresponding domain. As shown in Figure 4, performance improves rapidly with the first 1.5k-2k annotated examples, but plateaus after around 2.5k samples. This saturation suggests that early gains come from learning

general entity patterns, while additional examples offer diminishing returns due to redundancy or limited diversity in the unlabeled pool. Based on this trend, we find that annotating 1.5k–2.5k sentences is a practical and efficient choice for low-resource settings. Notably, with just 1k-1.5k annotated examples, our method already surpasses both ICL-MS and previous state-of-the-art methods. This demonstrates the data efficiency of our ICA framework and its ability to produce high-quality training data with minimal human supervision.

4.2 RQ2: How Much human-annotated Data is Needed for Effective ICA?

This experiment investigates how much human-annotated data is necessary to guide LLMs for ef-

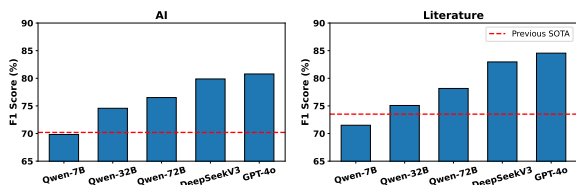


Figure 6: Comparison of LLMs on AI and Literature. The red dashed line indicates SOTA’s performance.

fective in-context annotation. Specifically, we vary the number of human-annotated examples settings used in the ICA prompt from 0 to 300. We then annotated 2k samples for each setting and used those data to train a BERT-based NER model.

As shown in Figure 5, performance improves significantly as the number of human-annotated examples increases from 0 to around 75–100. Beyond this point, gains become marginal, with nearly flat curves on both AI and Literature domains. These results suggest that using approximately 75–100 human-annotated examples is sufficient to enable strong ICA performance. Such a requirement is practically feasible, as collecting this amount of annotated data typically takes only a few hours. It strikes an effective balance between human effort and LLM utility for building high-quality NER training sets in low-resource domains.

4.3 RQ3: How Important is the Choice of LLM?

To study the impact of LLM choice on annotation quality, we conduct an experiment using different LLMs to perform in-context annotation. Specifically, each model is used to label 2k sentences for both the AI and Literature domains using our ICA framework with EAR. The annotated data is then used to train a BERT-based NER model. We compare several models of varying scale and architecture, including the frontier model **GPT-4o**, and three variants of **Qwen-2.5** with 7B, 32B, and 72B parameters. DeepSeekV3 is also included as a competitive baseline. As shown in Figure 6, the choice of LLM significantly impacts downstream performance. Larger and more capable models yield higher-quality annotations that translate into better NER performance. For both domains, GPT-4o achieves the best results, followed closely by DeepSeekV3 and Qwen-72B. In contrast, smaller models such as Qwen-7B lag behind by a wide margin, particularly on the more complex Literature domain. These results suggest that model scale and pretraining quality both matter for in-

context annotation. While open-source models can be competitive when scaled up (e.g., Qwen-72B), there remains a notable gap between smaller LLMs and the best proprietary models. This highlights a trade-off between annotation quality and model accessibility.

5 Related Work

Named Entity Recognitions. Large language models have been applied to NER via zero-shot and few-shot prompting, but early attempts fell short of the accuracy of task-specific models (Jimenez Gutierrez et al., 2022; Ashok and Lip-ton, 2023). Recent work focuses on closing this gap by better integrating task knowledge into prompts and more advanced prompt techniques (Zhang et al., 2025b). For example, PromptNER augments few-shot examples with explicit entity type definitions, prompting an LLM to list entities in a text along with explanations; Xie et al. (2023) proposes an inference strategy combining problem decomposition, syntactic augmentation, and a two-stage majority voting scheme, significantly improving ChatGPT’s zero-shot NER performance. Xie et al. (2024) involves self-annotation of unlabeled corpora, selection of reliable annotations, and in-context learning with retrieved demonstrations. Heng et al. (2024) introduces a cost-effective framework that guides large language models to self-reflect on domain-specific attributes and pre-generate entity terms, thereby creating high-quality NER datasets. Another batch approach is to design general-purpose NER systems inspired by LLMs’ flexibility. Methods such as GLiNER (Zaratiana et al., 2024), NuNER (Bogdanov et al., 2024), GliNER (Sainz et al., 2023), and UniversalNER (Zhou et al.) leverage large-scale synthetic NER data for extensive training, thereby achieving strong zero-shot performance. Another line of work builds unified NER frameworks to handle flat, nested, and discontinuous entities. UIE introduces a text-to-structure generation framework that unifies multiple IE tasks via schema-based prompting and a structured extraction language (Lu et al., 2022); LasUIE extends this by inducing latent syntactic structures and integrating them into the generation process (Fei et al., 2022). In the scientific domain, several corpora have been developed to support NER over scholarly texts, including DMDD for dataset mention detection (Pan et al., 2023), SciDMT for joint detection of datasets,

methods, and tasks (Pan et al., 2024a), and SciER for entity and relation extraction from full-text scientific publications (Zhang et al., 2024a). Since large-scale scientific corpora often rely on weak or distant supervision, label-cleaning approaches such as DynClean (Zhang et al., 2025a) have been proposed to improve annotation quality by leveraging training dynamics to identify mislabeled instances. However, these methods typically require substantial computational resources for pre-training and synthetic data generation. In contrast, our framework can serve as a cost-effective annotation generator, producing high-quality labels for training unified NER models.

Many-Shot In-Context Learning. Many studies suggested that using a small number of demonstrations, typically less than one hundred, was sufficient to achieve strong performance (Highmore, 2024; Min et al., 2022), and some even indicated that increasing the number of demonstrations could potentially hurt ICL performance (Chen et al., 2023). Recent long-context architectures expand context length by orders of magnitude (from a few thousand tokens in GPT-3 to around 1 million tokens in recent research models) (Agarwal et al., 2024), enabling many-shot in-context learning with hundreds or thousands of examples in a single prompt. Studies have observed that providing far more demonstrations often yields significant accuracy gains compared to the traditional few-shot regime (Agarwal et al., 2024; Bertsch et al., 2025). In fact, as the number of prompt examples grows into the hundreds, models begin to approach or even match fine-tuned performance on certain tasks (Agarwal et al., 2024), indicating that sufficiently large prompts allow LLMs to overcome some pre-training biases. One recent work Song et al. (2025) study whether many-shot ICL can improve the reliability of LLMs as evaluators, and finds that using many-shot prompts—especially with reference answers—leads to more consistent and accurate evaluation results. These findings demonstrate the feasibility of leveraging many-shot ICL to further enhance the performance of LLM-based ICL and to address the challenge of limited training data.

6 Conclusion

This work provides the first comprehensive investigation of many-shot in-context learning (ICL) for Named Entity Recognition (NER), demonstrating its viability as both an inference-time strategy and

a scalable annotation framework. We illustrate that with only a modest number of seed annotations (*around a hundred*), many-shot in-context annotation can generate high-quality labeled data, enabling significant downstream performance gains.

Building on these findings, we propose the ICA framework, which uses LLMs as offline annotators to efficiently generate and refine labeled data with minimal human effort. By decoupling LLM annotation from deployment, ICA enables compact models like BERT to be trained on high-quality data and deployed cost-effectively at scale. In low-resource settings, ICA achieves approximately 10% absolute F1 improvement over previous SOTA methods. Our findings highlight the promise of LLM-based ICL and annotation frameworks like ICA in improving data efficiency and reducing annotation costs for NER and related tasks.

Limitations

While our study demonstrates the promise of many-shot in-context learning (ICL) for NER, several limitations remain. While our method reduces the need for large-scale manual annotation, it still requires a moderate number of high-quality seed labels and introduces computational overhead from repeated LLM inference. In addition, ICL offers little transparency into why certain outputs are generated or how LLMs adapt to increasing demonstration size, complicating efforts to debug or audit model behavior in sensitive applications. Finally, our experiments focus on English-language NER in a limited set of domains. Further evaluation is needed to understand the generalizability of many-shot ICL to other languages, domains, and entity types, especially those with more complex or ambiguous annotation schemes. Future work can address these limitations by exploring hybrid models that combine the efficiency of ICL with targeted fine-tuning, extending evaluations to multilingual and multi-domain settings.

Acknowledgements

This work was supported by the National Science Foundation awards III-2107213, III-2107518, 2026513, and ITE-2333789. We thank the reviewers and the area chair for their constructive feedback and suggestions, which have substantially improved the quality of this paper.

References

- Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, and 1 others. 2024. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966.
- Dhananjay Ashok and Zachary C Lipton. 2023. Prompter: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.
- Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. 2025. In-context learning with long-context models: An in-depth exploration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12119–12149, Albuquerque, New Mexico. Association for Computational Linguistics.
- Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne P Bernard. 2024. NuNER: Entity recognition encoder pre-training via LLM-annotated data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11829–11841, Miami, Florida, USA. Association for Computational Linguistics.
- 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.
- Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. 2023. How many demonstrations do you need for in-context learning? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11149–11159, Singapore. Association for Computational Linguistics.
- Shivani Choudhary, Niladri Chatterjee, and Subir Saha. 2023. IITD at SemEval-2023 task 2: A multi-stage information retrieval approach for fine-grained named entity recognition. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 800–806, Toronto, Canada. Association for Computational Linguistics.
- Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- DeepSeek. 2024. Introducing deepseek-v3.
- Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gestein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.
- 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, Copenhagen, Denmark. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.
- Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Krungkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. Association for Computational Linguistics.
- Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.
- 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 *ACL*, pages 3198–3213.
- Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Yan Bowen, and Min Zhang. 2024. Rethinking negative instances for generative named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3461–3475, Bangkok, Thailand. Association for Computational Linguistics.
- Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. *Advances in Neural Information Processing Systems*, 35:15460–15475.
- Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. In *Proceedings of the 41st International Conference on Machine Learning*, pages 14125–14134.

- Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Yifan Hao, Xingyuan Pan, Hanning Zhang, Chenlu Ye, Rui Pan, and Tong Zhang. 2025. Understanding over-adaptation in supervised fine-tuning: The role of ensemble methods. *arXiv preprint arXiv:2506.01901*.
- 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*, pages 15992–16030, Bangkok, Thailand. Association for Computational Linguistics.
- Clyde Highmore. 2024. In-context learning in large language models: A comprehensive survey. *Preprints*, 202407:v1.
- Jinpeng Hu, He Zhao, Dan Guo, Xiang Wan, and Tsung-Hui Chang. 2022. [A label-aware autoregressive framework for cross-domain NER](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2222–2232, Seattle, United States. Association for Computational Linguistics.
- Xuming Hu, Zhaochen Hong, Yong Jiang, Zhichao Lin, Xiaobin Wang, Pengjun Xie, and Philip S Yu. 2024a. Three heads are better than one: improving cross-domain ner with progressive decomposed network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18261–18269.
- Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, and 1 others. 2024b. Improving large language models for clinical named entity recognition via prompt engineering. *Journal of the American Medical Informatics Association*, 31(9):1812–1820.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.
- Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. [Thinking about GPT-3 in-context learning for biomedical IE? think again](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.
- Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. [When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs](#). *Transactions of the Association for Computational Linguistics*, 12:1417–1440.
- 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.
- Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13452–13460.
- Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. *arXiv preprint arXiv:2203.12277*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.
- Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. [Named entity recognition with partially annotated training data](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 645–655, Hong Kong, China. Association for Computational Linguistics.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- 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: Industry Track*, pages 686–696, Miami, Florida, US. Association for Computational Linguistics.
- OpenAI. 2024. [Hello gpt-4](#).
- Huitong Pan, Mustapha Adamu, Qi Zhang, Eduard Dragut, and Longin Jan Latecki. 2025a. [ClimateIE](#):

- A dataset for climate science information extraction. In *Proceedings of the 2nd Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2025)*, pages 76–98, Vienna, Austria. Association for Computational Linguistics.
- Huitong Pan, Qi Zhang, Mustapha Adamu, Eduard Dragut, and Longin Jan Latecki. 2025b. **Taxonomy-driven knowledge graph construction for domain-specific scientific applications**. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4295–4320, Vienna, Austria. Association for Computational Linguistics.
- Huitong Pan, Qi Zhang, Cornelia Caragea, Eduard Dragut, and Longin Jan Latecki. 2024a. **SciDMT: A large-scale corpus for detecting scientific mentions**. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14407–14417, Torino, Italia. ELRA and ICCL.
- Huitong Pan, Qi Zhang, Eduard Dragut, Cornelia Caragea, and Longin Jan Latecki. 2023. **DMDD: A large-scale dataset for dataset mentions detection**. *Transactions of the Association for Computational Linguistics*, 11:1132–1146.
- Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024b. **Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies**. *Transactions of the Association for Computational Linguistics*, 12:484–506.
- Reid Pryzant, Ziyi Yang, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. **Automatic rule induction for efficient semi-supervised learning**. In *Findings of the EMNLP*, pages 28–44.
- Esteban Safranchik, Shiyang Luo, and Stephen Bach. 2020. **Weakly supervised sequence tagging from noisy rules**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5570–5578.
- Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. **Gollie: Annotation guidelines improve zero-shot information-extraction**. *arXiv preprint arXiv:2310.03668*.
- Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. **Learning named entity tagger using domain-specific dictionary**. In *EMNLP*, pages 2054–2064.
- Amanpreet Singh, Mike D’Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. **SciRepEval: A multi-format benchmark for scientific document representations**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore. Association for Computational Linguistics.
- Mingyang Song, Mao Zheng, and Xuan Luo. 2025. **Can many-shot in-context learning help LLMs as evaluators? a preliminary empirical study**. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8232–8241, Abu Dhabi, UAE. Association for Computational Linguistics.
- Mo Sun, Siheng Xiong, Yuankai Cai, and Bowen Zuo. 2025. **Positional attention for efficient bert-based named entity recognition**. *arXiv preprint arXiv:2505.01868*.
- Minghao Tang, Peng Zhang, Yongquan He, Yongxiu Xu, Chengpeng Chao, and Hongbo Xu. 2022. **DoSEA: A domain-specific entity-aware framework for cross-domain named entity recognition**. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2147–2156, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Erik F. 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 at HLT-NAACL 2003*, pages 142–147.
- Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. **GPT-NER: Named entity recognition via large language models**. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275, Albuquerque, New Mexico. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. **Self-consistency improves chain of thought reasoning in language models**. In *The Eleventh International Conference on Learning Representations*.
- Xiang Wei, Yufeng Chen, Ning Cheng, Xingyu Cui, Jinan Xu, and Wenjuan Han. 2024. **CollabKG: A learnable human-machine-cooperative information extraction toolkit for (event) knowledge graph construction**. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3490–3506, Torino, Italia. ELRA and ICCL.
- 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, Singapore. Association for Computational Linguistics.
- Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. **Self-improving for zero-shot named entity recognition with large language models**. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*

- (Volume 2: Short Papers), pages 583–593, Mexico City, Mexico. Association for Computational Linguistics.
- Jingyun Xu and Yi Cai. 2023. Decoupled hyperbolic graph attention network for cross-domain named entity recognition. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 591–600.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Yuming Yang, Wantong Zhao, Caishuang Huang, Junjie Ye, Xiao Wang, Huiyuan Zheng, Yang Nan, Yuran Wang, Xueying Xu, Kaixin Huang, Yunke Zhang, Tao Gui, Qi Zhang, and Xuanjing Huang. 2025b. Beyond boundaries: Learning a universal entity taxonomy across datasets and languages for open named entity recognition. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10902–10923, Abu Dhabi, UAE. Association for Computational Linguistics.
- Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.
- Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. 2024a. SciER: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13083–13100, Miami, Florida, USA. Association for Computational Linguistics.
- Qi Zhang, Huitong Pan, Zhijia Chen, Longin Jan Latecki, Cornelia Caragea, and Eduard Dragut. 2025a. DynClean: Training dynamics-based label cleaning for distantly-supervised named entity recognition. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2540–2556, Albuquerque, New Mexico. Association for Computational Linguistics.
- Xinghua Zhang, Bowen Yu, Xin Cong, Taoyu Su, Quangan Li, Tingwen Liu, and Hongbo Xu. 2024b. Cross-domain ner under a divide-and-transfer paradigm. *ACM Transactions on Information Systems*, 42(5):1–32.
- Xinghua Zhang, Bowen Yu, Yubin Wang, Tingwen Liu, Taoyu Su, and Hongbo Xu. 2022. Exploring modular task decomposition in cross-domain named entity recognition. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 301–311.
- Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. 2025b. A survey of generative information extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4840–4870, Abu Dhabi, UAE. Association for Computational Linguistics.
- Junhao Zheng, Haibin Chen, and Qianli Ma. 2022. Cross-domain named entity recognition via graph matching. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2670–2680, Dublin, Ireland. Association for Computational Linguistics.
- Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

A Additional Details of Many-Shot ICL Study

A.1 Dataset Statistics

We consider 4 NER datasets from different domains: CoNLL2003 (Tjong Kim Sang and De Meulder, 2003), MIT-Movie and MIT-Restaurant (Liu et al., 2013), WNUT 2017 (Derczynski et al., 2017).

Name	Domain	Entity Types	Data size
CoNLL 2003	News	4	14,041 / 3,250 / 3,453
WNUT 2017	SNS	6	1,000 / 1,008 / 1,287
MIT Restaurant	Restaurant review	8	7,660 / - / 1,521
MIT Movie	Movie review	12	7,816 / - / 1,953

Table 2: Four NER datasets are considered. Data size is the number of sentences in training/validation/test set.

A.2 Many-shot ICL Models and Settings Details

LLMs details. The majority of the analysis in the paper concerns these five LLMs models:

- **GPT-4o** (OpenAI, 2024) is a recent multi-modal model developed by OpenAI, designed for fast and cost-efficient inference. It supports long-context inputs and demonstrates strong reasoning capabilities.
- **DeepSeekV3** (DeepSeek, 2024) is a decoder-only language model optimized for both efficiency and long-context understanding.
- **Qwen2.5-7B and Qwen2.5-72B** (Yang et al., 2025a) are multilingual decoder-only models with trained context lengths of 128k and 131k tokens, respectively. We use their instruction-tuned variants in our experiments.
- **LLaMA3.1-8B and LLaMA3.1-70B** (Grattafiori et al., 2024) are the latest models in Meta’s LLaMA series, featuring improvements in both training data and model architecture. Both variants support a trained context length of 128k tokens. We employ the instruction-tuned versions in all evaluations.

We use the vLLM² v0.8.5 to deploy the Qwen2.5 and LLaMa 3.1 LLMs on a machine with 4 A100

²<https://docs.vllm.ai/en/stable/>

GPUs. For the DeepSeekV3 and GPT-4o, we use the official provided API. All the experiments results are obtained during the January 20, 2025 to May 10, 2025.

BERT Fine-Tuning details. All BERT-FT models in this paper refer to the fine-tuned *bert-base-cased*³ version. We use the same combination of hyperparameters for all datasets: the learning rate is set as 1e-5; the training batch size is 16, all results are based on the fine-tuned results of 5 epochs and report average of five runs, each with a different random seed. Experiments were conducted using a single A100 GPU card with PyTorch 2.7.0. The average running time on the CoNLL03 dataset is 34 seconds/epoch.

A.3 Baseline Implementation Details

ProgGen. We implemented ProgGen (Heng et al., 2024) on CrossNER by following the authors’ released code and recipe for data generation and training. We adopted the best-performing configuration reported in the paper—combining their data generation and refinement pipeline with the strongest downstream model setup—and ran all experiments under this setting to ensure comparability with their reported results.

GliNER. We fine-tuned the GliNER encoder (Zaratiana et al., 2024) on CrossNER using 100 human-annotated examples per domain to match the data budget of our ICA framework.

Other Baselines. For the remaining baselines, we report results on CrossNER directly from the original sources for fair comparison. Note that many of these baselines use not only the 100 human-annotated training examples but also additional validation data for hyperparameter tuning.

A.4 Detail Many-shot ICL for NER results

Table 3 provides the complete performance results of all seven language models under both random and retrieval-based ICL settings across the four datasets. Figure 7 and 8 presents the performance trends of all models as the number of in-context examples increases.

From the figures, we observe a consistent trend: the performance of LLMs generally improves as the number of in-context examples increases, especially under the random sampling setting. For smaller models such as Qwen2.5-7B and

³<https://huggingface.co/bert-base-cased>

LLaMA3.1-8B, although their final performance may not match that of the BERT-FT baseline, they still exhibit a clear many-shot ICL improvement trend as the number of demonstrations grows.

B Additional Details of ICA Framework

B.1 More Detail of Annotation Refinement Methods

In this section, we provide detailed descriptions of the three refinement strategies used to enhance the quality of LLM-generated annotations: self-consistency §B.1.1, self-correction §B.1.2, and error-aware refinement §B.1.3.

B.1.1 Self-consistency

To improve the robustness of LLM-generated annotations, we adopt a self-consistency refinement strategy inspired by prior work on decoding via multiple sampling (Wang et al.; Xie et al., 2024). In our NER setting, we apply self-consistency to low-confidence samples as identified by the confidence scoring method in Section 3.1.

For each selected sentence, we prompt the LLM to generate three independent predictions by randomly varying the order of in-context demonstration examples in the prompt. This results in three separate XML-style annotations for the same sentence.

We then perform majority voting over the predicted entities at both the span and type levels:

- **Span-level voting:** A candidate span is retained only if it appears in at least two out of the three predictions.
- **Type-level voting:** For spans that appear in multiple predictions, i.e., appear at least two times, the entity type assigned by the majority is selected as the final label.

This voting process filters out inconsistent or spurious predictions and prioritizes entities with high agreement across multiple completions. The final refined annotation includes only those entity mentions that meet the self-consistency threshold at both the span and type level.

B.1.2 Self-correction

Self-correction is a lightweight refinement strategy that leverages the LLM’s own reasoning ability to revise and improve its initial predictions (Pan et al., 2024b; Kamoi et al., 2024; Heng et al., 2024; Madaan et al., 2023). Given an initial annotation

from ICA, we construct a second prompt that asks the LLM to review its output and make corrections if any errors are detected.

To help the LLM learn this behavior, we employ a few-shot prompting strategy. Specifically, we begin by randomly sampling 5 sentences from the 100 human-labeled seed examples and holding them out. We then use the LLM to generate predictions for the remaining 95 sentences. From these predictions, we identify 6 examples that contain typical NER errors, two each for **spurious entity**, **missing entity**, and **type error**, as well as 4 examples that are correctly annotated. Each selected example is paired with its corrected version and included in the prompt as a demonstration. The other predictions are discarded and not used for prompting.

Figure 9 shows the structure of our self-correction prompt, which consists of four parts: a brief description of common NER error types, entity type definitions, 10 few-shot examples (6 erroneous and 4 correct), and the target sentence with its current and refined annotations.

We do not explicitly target span boundary errors, as partially correct entity spans have been shown to be useful for training NER models (Mayhew et al., 2019).

B.1.3 Error-Aware Refinement

Error-aware refinement (EAR) is a targeted strategy that applies a series of error-specific prompts to improve the quality of LLM-generated annotations. Unlike self-correction (Appendix B.1.2), which uses a single prompt to handle all types of errors, EAR decomposes the refinement process into three independent steps, each addressing one specific error type: spurious entities, missing entities, or incorrect types.

Each step is handled using a dedicated prompt, and the three prompts are applied sequentially to the same annotated sentence. For example, we first remove spurious entities, then add missing ones, and finally correct entity types. This modular setup improves controllability, avoids conflicting operations, and enables precise error handling.

To help the LLM learn how to correct specific error types, we follow the same few-shot example construction strategy used in the self-correction setup. We identify 10 examples from LLM-generated predictions—six containing typical NER errors (two per error type) and four that are correctly annotated. Each selected example is

	MIT-Movie			MIT-Res			CoNLL2003			WNUT2017					
	# Shots	P	R	F1	P	R	F1	P	R	F1	P	R	F1		
SOTA	-	-	-	71.20	-	-	-	79.60	-	-	-	94.30	-	-	52.80
BERT	58.92	63.80	61.17	76.33	81.43	78.77	86.79	89.12	87.92	-	-	-	-	50.30	
Qwen-72B	0	46.20	34.70	39.63	56.94	46.19	51.01	64.37	60.61	62.43	30.16	49.44	37.46		
	5	56.18	60.73	58.37	64.77	67.88	66.29	65.13	73.59	69.11	33.87	62.45	43.92		
	10	58.62	62.90	60.68	74.96	79.71	66.61	71.08	76.62	73.75	44.03	52.04	47.70		
	25	63.28	66.10	64.66	65.52	69.21	67.31	77.10	73.59	75.30	46.34	49.44	47.84		
	50	63.21	69.63	66.27	68.73	73.51	71.04	80.63	86.09	83.27	44.51	55.76	49.50		
	100	64.28	68.61	66.37	72.13	75.00	73.54	82.99	86.58	84.75	47.50	56.51	51.61		
	200	64.71	69.29	66.92	74.39	75.99	75.18	84.89	86.36	85.62	45.65	56.51	50.50		
	300	63.84	70.55	67.03	75.20	78.31	76.72	86.57	90.69	88.58	48.32	58.74	53.02		
	400	64.68	69.57	67.03	74.66	80.96	77.68	87.99	91.99	89.95	54.55	53.53	54.03		
	500	65.51	70.89	68.09	77.98	82.03	79.96	88.89	92.17	90.50	46.59	67.96	55.28		
Qwen-72B (BM25)	5	62.89	67.12	64.94	67.23	72.35	69.70	76.16	80.51	78.27	37.73	60.59	46.50		
	10	62.64	69.29	65.80	71.36	76.32	73.76	82.92	86.15	84.50	39.67	54.28	45.84		
	25	63.49	69.29	66.27	73.39	79.47	76.31	86.19	89.18	87.66	43.34	52.04	47.30		
	50	66.78	68.84	67.79	73.75	80.46	76.96	85.68	89.39	87.50	43.32	54.28	48.18		
	100	67.88	69.98	68.92	74.09	80.96	77.37	88.21	90.69	89.43	48.23	50.56	49.36		
	200	67.58	70.43	68.98	76.24	80.35	78.24	87.63	90.48	89.03	48.34	54.28	51.14		
	300	67.54	71.00	69.23	76.07	82.62	79.21	87.76	91.56	89.62	52.00	53.16	52.57		
	400	68.10	72.37	70.17	77.69	81.14	79.38	89.11	91.84	90.45	57.96	52.79	55.25		
	500	71.07	71.67	71.37	77.64	82.78	80.13	89.95	92.27	91.10	50.23	60.77	55.00		
	DeepSeekV3	0	51.67	56.39	53.93	56.98	46.59	51.26	61.40	68.18	64.62	39.52	55.39	46.13	
5		58.33	55.26	56.76	64.80	64.18	64.49	64.46	75.76	69.65	39.05	60.97	47.61		
10		65.15	67.24	66.18	65.45	65.45	65.45	76.77	82.25	79.41	53.13	50.56	51.81		
25		64.79	69.75	67.18	66.77	70.36	68.52	77.41	81.60	79.45	52.19	53.16	52.67		
50		64.76	69.86	67.22	73.04	76.86	74.90	79.50	83.12	81.27	53.33	53.53	53.43		
100		67.17	71.46	69.25	74.09	77.50	75.76	84.39	88.96	86.62	56.38	50.93	53.52		
200		68.32	72.37	70.29	74.02	78.13	76.02	87.47	87.66	87.57	52.25	56.13	54.12		
300		67.97	73.17	70.48	74.66	79.40	76.96	87.08	88.96	88.01	60.75	48.33	53.83		
400		68.57	73.97	71.17	75.84	78.61	77.20	87.29	89.18	88.22	60.52	52.42	56.18		
500		70.39	74.09	72.19	76.44	81.78	79.02	87.29	89.18	88.22	56.91	59.12	57.99		
DeepSeekV3 (BM25)	5	66.17	71.69	68.82	69.77	76.07	72.78	80.80	87.45	83.99	41.41	58.56	48.51		
	10	67.38	72.37	69.79	74.11	79.40	76.66	87.90	89.61	88.75	51.04	54.14	52.55		
	25	67.91	72.95	70.34	75.08	79.24	77.10	88.91	90.26	89.58	54.65	52.42	53.51		
	50	67.71	74.43	70.91	77.36	77.97	77.66	89.01	91.13	90.05	57.03	52.79	54.83		
	100	68.34	74.43	71.26	76.91	81.30	79.04	89.01	91.13	90.05	48.29	62.43	54.46		
	200	69.63	75.91	72.64	77.79	81.62	79.66	89.08	91.77	90.41	56.50	55.25	55.87		
	300	68.87	76.03	72.27	77.58	82.25	79.85	89.19	91.13	90.15	58.23	53.90	55.98		
	400	69.82	76.60	73.05	77.89	82.09	79.94	90.06	92.21	91.12	54.77	60.22	57.37		
	500	70.35	76.37	73.23	77.61	82.41	79.94	90.59	91.50	91.04	55.73	59.12	57.37		
	GPT-4o	0	58.18	60.50	59.32	63.71	53.41	58.10	62.71	73.16	67.53	39.42	60.97	47.88	
5		62.10	65.64	63.82	64.82	68.62	66.67	77.04	75.54	76.28	45.58	61.34	52.30		
10		63.12	64.72	63.91	66.62	70.52	68.51	79.70	81.60	80.64	48.14	57.62	52.45		
25		63.15	66.32	64.70	68.37	69.89	69.12	83.76	84.85	84.30	50.99	57.25	53.94		
50		63.31	68.15	65.64	69.42	71.95	70.66	87.36	86.80	87.08	47.47	62.83	54.08		
100		63.14	68.84	65.87	70.31	74.33	72.27	86.94	87.88	87.41	51.13	59.11	54.83		
200		63.59	69.98	66.63	71.69	74.64	73.14	86.32	88.74	87.51	52.00	57.99	54.83		
300		65.34	69.76	67.48	74.45	80.86	77.52	86.68	88.74	87.70	54.33	58.36	56.27		
400		68.32	72.37	70.29	74.51	81.82	77.99	86.82	89.83	88.30	53.59	60.97	57.04		
500		69.84	74.54	72.11	76.50	82.54	79.40	88.94	88.74	88.84	53.33	62.45	57.53		
GPT-4o (BM25)	5	64.83	68.38	66.56	71.54	74.48	72.98	68.29	90.91	77.99	54.09	56.51	55.27		
	10	66.41	70.21	68.26	70.92	75.75	73.26	83.44	88.31	85.80	52.33	58.58	55.28		
	25	66.74	70.78	68.70	71.24	75.75	73.43	86.02	87.88	86.94	49.56	62.83	55.41		
	50	67.49	71.35	69.37	71.01	76.07	73.45	87.78	91.77	89.74	52.27	59.85	55.81		
	100	68.72	71.23	69.96	73.66	79.33	76.39	87.97	91.77	89.83	52.27	61.22	56.39		
	200	68.85	72.15	70.46	75.29	79.09	77.14	89.70	90.48	90.09	55.96	57.84	56.88		
	300	69.79	73.06	71.39	75.96	80.53	78.18	88.54	91.99	90.23	54.15	60.59	57.19		
	400	70.19	74.20	72.14	75.85	80.77	78.23	90.73	91.32	91.03	54.05	62.08	57.79		
	500	70.57	75.34	72.88	77.63	81.73	79.63	89.89	92.42	91.14	57.25	61.24	59.18		
	Llama3.1-70B	0	43.75	41.18	42.42	43.37	61.30	50.80	40.51	66.67	50.39	26.36	37.57	30.98	
5		46.09	56.94	50.94	54.79	63.22	58.71	66.24	67.97	67.09	26.17	46.41	33.47		
10		49.02	58.14	53.19	56.67	65.38	60.71	59.02	84.03	69.34	30.72	49.72	37.97		
25		52.73	61.70	56.86	58.73	67.27	62.71	60.61	89.24	72.19	38.30	48.65	42.86		
50		54.03	63.62	58.43	56.91	59.12	66.06	65.44	80.74	72.29	41.67	52.63	46.51		
100		62.75	65.73	64.21	70.26	70.43	70.35	77.87	80.74	79.28	43.84	56.88	49.51		
200		63.08	66.96	64.96	70.90	72.58	71.73	76.31	86.11	80.91	46.43	56.52	50.98		
300		64.68	66.61	65.63	70.93	74.64	72.74	79.17	85.76	82.33	48.03	54.28	50.96		
400		64.52	66.78	65.63	70.83	75.44	73.06	83.55	88.19	85.81	50.00	53.53	51.71		
500		67.35	69.60	68.45	72.47	77.18	74.75	87.21	92.36	89.71	49.17	55.39	52.10		
Llama3.1-70B (BM25)	5	58.18	66.26	61.96	48.99	75.48	59.41	55.31	86.81	67.57	31.52	44.75	36.99		
	10	61.22	65.22	63.16	54.56	77.64	64.09	65.79	86.81	74.85	38.78	48.72	43.18		
	25	63.95	68.89	66.33	66.67	72.00	69.23	65.45	88.96	75.41	42.22	51.35	46.34		
	50	66.00	67.35	66.67	69.09	74.51	71.70	79.88	89.58	84.45	44.68	53.85	48.84		
	100	66.45	71.35	68.81	71.59	75.72	73.60	81.78	89.39	85.42	42.82	58.74	49.53		
	200	66.72	71.53	69.04	71.85	76.86	74.27	84.39	92.01	88.04	52.34	49.81	51.05		
	300	67.49	71.88	69.62	72.54	77.02	74.71	84.54	93.06	88.60	50.00	52.79	51.36		
	400	66.88	72.76	69.70	72.63	77.81	75.13	85.85	92.71	89.15	52.04	52.04	52.04		
	500	67.26	73.64	70.30	72.69	78.45	75.46	89.33	92.42	90.85	53.01	52.42	52.71		
	Qwen2.5-7B	0	35.40	32.16	33.70	46.59	27.88	34.89	47.28	39.24	42.88	57.69	8.29	14.49	
5		51.72	52.90	52.30	46.51	48.08	47.28	66.30	42.36	51.69	59.38	10.50	17.84		
10		63.38	50.79	56.39	54.52	52.16	53.32	59.50	50.00	54.34	54.05	11.05	18.35		
25		60.27	54.66	57.33	57.14	51.92									

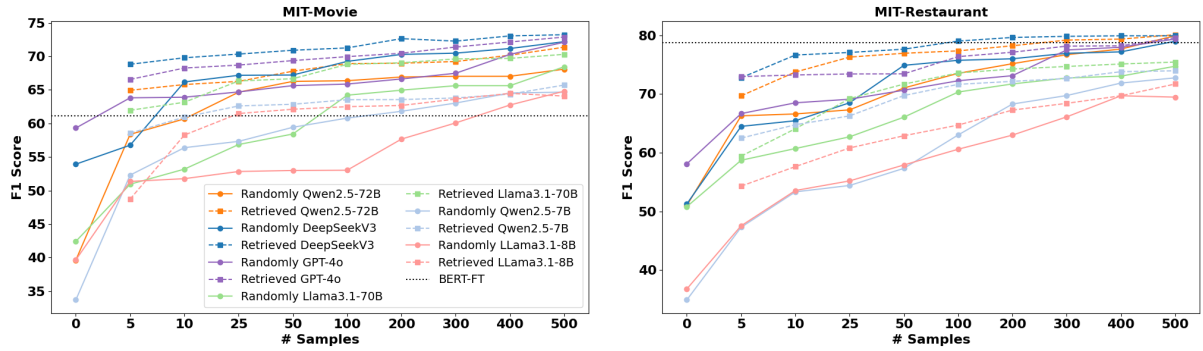


Figure 7: Complete performance results on the MIT-Movie and MIT-Restaurant datasets as the number of samples increases.

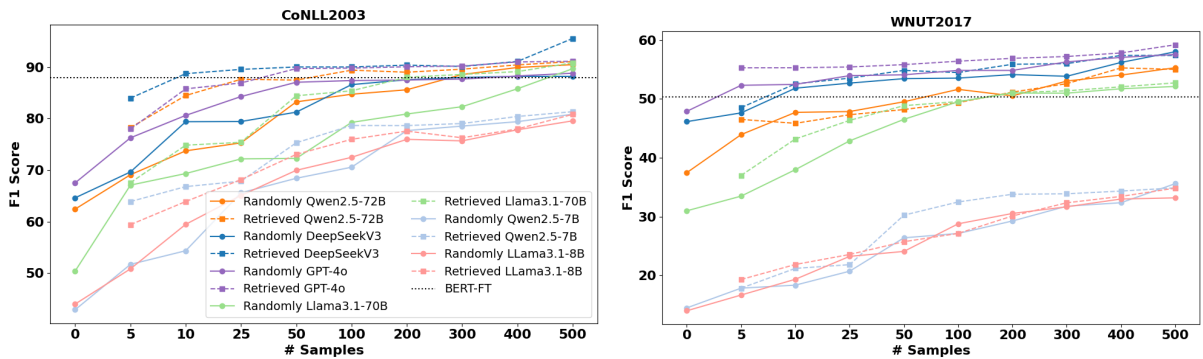


Figure 8: Complete performance results on the CoNLL2003 and WNUT2017 datasets as the number of samples increases.

paired with its corrected version to form a demonstration pair.

To construct error-specific prompts, we further process these examples to isolate the target error type: for each prompt, we retain only the target error in the example and pre-correct all other types of errors within the same sentence. This ensures that each demonstration focuses solely on the intended error type, enabling the LLM to learn a precise and unambiguous correction behavior.

Unlike self-correction, where all examples are combined into a single prompt, EAR builds three separate prompts—each containing only examples of one specific error type. This design allows the LLM to focus on a single correction behavior at a time, without being distracted by unrelated errors.

Figures 10, 11, and 12 illustrate the structure of the three error-specific prompts used in this step-by-step refinement process.

B.2 CrossNER benchmark

Table 4 provided the detail statistics of each domain in CrossNER benchmark.

B.3 Case Study

To better understand the effect of our proposed self-refinement methods, we conduct a qualitative analysis by reviewing several annotated samples from the AI domains. As illustrative examples, we present two representative cases in Figures 13 and 14.

In Figure 13, we observe that the initial ICA annotation incorrectly classifies “1 regression analysis” as a field entity, whereas it should be labeled as a task. All three refinement strategies successfully correct this type error. Additionally, EAR identifies an extra missing entity—“squared deviations”—and correctly labels it as miscellaneous, which was not detected by ICA.

In Figure 14, all three methods correct the misclassification of “data-mining”, which ICA originally labeled incorrectly, assigning it the correct type field. They also successfully remove “intelligence”, which was a spurious entity in the original annotation. Moreover, EAR further removes “high-level ontologies”, another spurious entity mistakenly included by ICA, demonstrating its stronger ability to filter over-predicted spans.

Domain	Unlabeled Corpus	Labeled NER			Entity Categories
	# sentence	# Train	# Dev	# Test	
Reuters	-	14,987	3,466	3,684	person, organization, location, miscellaneous
Politics	9.07M	200	541	651	politician, person, organization, political party, event, election, country, location, miscellaneous
Natural Science	5.32M	200	450	543	scientist, person, university, organization, country, location, discipline, enzyme, protein, chemical compound, chemical element, event, astronomical object, academic journal, award, theory, miscellaneous
Music	9.82M	100	380	456	music genre, song, band, album, musical artist, musical instrument, award, event, country, location, organization, person, miscellaneous
Literature	9.17M	100	400	416	book, writer, award, poem, event, magazine, person, location, organization, country, miscellaneous
Artificial Intelligence	287.62K	100	350	431	field, task, product, algorithm, researcher, metrics, university, country, person, organization, location, miscellaneous

Table 4: Data statistics of unlabeled domain corpora, labeled NER samples and entity categories for each domain.

Task: You are a careful NER annotation assistant. Below are several examples where the model's initial annotation either contains errors or is correct, followed by the corresponding refined version. Your task is to read a new sentence and its current annotation, then determine whether a correction is needed. Only make changes if you are confident. If the current annotation is already correct, output exactly: "All annotations are correct".

Error Types:

- Spurious Entity: an entity is marked but should not be. You must remove such entities if present.
- Missing Entity: a valid entity in the sentence is not marked. You must add such entities if found.
- Type Error: the entity is labeled with the wrong type. You must correct the type if necessary.

Entity Type Definitions:
{entity_list}

Examples:
{examples}

Sentence:
Current Annotation:
Refined Annotation:

Figure 9: Illustration of our self-correction prompt format.

B.4 Qualitative Observations and Design Motivations

Beyond the quantitative results, we encountered several recurring challenges during the development of our ICA framework. These observations directly motivated the design choices behind our Error-Aware Refinement (EAR) strategy.

Sycophancy in Generic Self-Correction. Our initial attempts used a generic prompt asking the LLM to “check and fix any errors.” We observed that the model frequently over-corrected its own outputs—deleting valid entities or altering correct types simply because it was prompted to look for

Task: You are a careful NER annotation assistant. Below are several examples where the model's initial annotation contains spurious entities, followed by the corrected version. Your job is to review a new sentence and its current annotation. If any entity is incorrectly marked and should not be annotated, remove it. Only remove entities if you are confident. If all annotations are correct, output exactly: "All annotations are correct".

Entity Type Definitions:
{entity_list}

Examples:
{examples}

Sentence:
Current Annotation:
Refined Annotation:

Figure 10: Prompt format for spurious entity error refinement.

mistakes—resulting in a notable drop in recall. This behavior is consistent with recent findings on the limitations of intrinsic self-correction (Huang et al., 2023; Hao et al., 2025). The instability of open-ended correction directly motivated our decision to decompose refinement into three specific, directed sub-tasks in EAR (spurious, missing, type), where each prompt focuses on a single, well-defined error pattern.

Over-Sensitivity to Capitalization. We found that LLMs tend to be over-sensitive to surface-level features such as capitalization. Non-entity capitalized phrases (e.g., “Table 1,” “Section 3,” or generic references like “The University”) were frequently tagged as named entities. This failure mode was persistent across model sizes and motivated the dedicated Spurious-Entity step in our pipeline, which

Task: You are a careful NER annotation assistant. Below are several examples where the model's initial annotation misses one or more valid entities, followed by the corrected version. Your job is to review a new sentence and its current annotation. If any valid entity is missing, add it using the correct type. Only add entities if you are confident. If all annotations are correct, output exactly: "All annotations are correct".

Entity Type Definitions:
{entity_list}

Examples:
{examples}

Sentence:
Current Annotation:
Refined Annotation:

Figure 11: Prompt format for missing entity recovery.

Task: You are a careful NER annotation assistant. Below are several examples where an entity is labeled with the wrong type, followed by the corrected version. Your job is to review a new sentence and its current annotation. If any entity is labeled with an incorrect type, correct it. Only correct entity types if you are confident. If all annotations are correct, output exactly: "All annotations are correct".

Entity Type Definitions:
{entity_list}

Examples:
{examples}

Sentence:
Current Annotation:
Refined Annotation:

Figure 12: Prompt format for type error correction.

specifically targets this type of false positive.

Boundary Ambiguity in Specialized Domains.

In the Science and Literature domains, models struggled with precise span boundaries—for example, whether to include determiners or long modifiers in a book title or a chemical compound name. We found that expanding the number of in-context demonstrations mitigated this issue more effectively than adjusting zero-shot instructions, which aligns with our main finding on the benefit of scaling demonstrations.

B.5 Cost Analysis and Deployment Scalability of ICA

Here we provide additional Analysis regarding the inference cost and scalability of our proposed ICA method. As explicitly discussed in Section B, our

Sentence: Also in regression analysis , mean squared error , often referred to as mean squared prediction error or out-of-sample mean squared error , can refer to the mean value of the squared deviations of the predictions from the TRUE values , over an out-of-sample test space , generated by a model estimated over a particular sample space .

Target Annotation: Also in <entity type="task">regression analysis</entity> , <entity type="metrics">mean squared error</entity> , often referred to as <entity type="metrics">mean squared prediction error</entity> or <entity type="metrics">out-of-sample mean squared error</entity> , can refer to the mean value of the <entity type="miscellaneous">squared deviations</entity> of the predictions from the TRUE values , over an out-of-sample test space , generated by a model estimated over a particular sample space .

ICA: Also in <entity type="field">regression analysis</entity> , <entity type="metrics">mean squared error</entity> , often referred to as <entity type="metrics">mean squared prediction error</entity> or <entity type="metrics">out-of-sample mean squared error</entity> , can refer to the mean value of the squared deviations of the predictions from the TRUE values , over an out-of-sample test space , generated by a model estimated over a particular sample space .

ICA w/ Self-consistency: Also in <entity type="task">regression analysis</entity> , <entity type="metrics">mean squared error</entity> , often referred to as <entity type="metrics">mean squared prediction error</entity> or <entity type="metrics">out-of-sample mean squared error</entity> , can refer to the mean value of the squared deviations of the predictions from the TRUE values , over an out-of-sample test space , generated by a model estimated over a particular sample space .

ICA w/ Self-correction: Also in <entity type="task">regression analysis</entity> , <entity type="metrics">mean squared error</entity> , often referred to as <entity type="metrics">mean squared prediction error</entity> or <entity type="metrics">out-of-sample mean squared error</entity> , can refer to the mean value of the squared deviations of the predictions from the TRUE values , over an out-of-sample test space , generated by a model estimated over a particular sample space .

ICA w/ EAR: Also in <entity type="task">regression analysis</entity> , <entity type="metrics">mean squared error</entity> , often referred to as <entity type="metrics">mean squared prediction error</entity> or <entity type="metrics">out-of-sample mean squared error</entity> , can refer to the mean value of the <entity type="miscellaneous">squared deviations</entity> of the predictions from the TRUE values , over an out-of-sample test space , generated by a model estimated over a particular sample space .

Figure 13: Case study from the AI dataset illustrating how the self-refinement methods correct annotation, where underline means the refinement method correct the error successfully.

framework leverages LLMs exclusively as offline annotators rather than as real-time inference engines. The annotation process involving LLMs is a one-time cost incurred offline. After this initial annotation phase, subsequent inference and practical deployment rely solely on compact and efficient small language models (e.g., BERT), significantly reducing inference costs and enabling highly scalable deployments.

To further illustrate this benefit, we provide a concrete cost analysis based on reported annotation costs from prior literature. According to (Ding et al., 2023), human annotation for the AI domain incurs a cost of \$42.85 per 100 samples. Consequently, the previous state-of-the-art method (DTrans-MPrompt), which requires 450 human-labeled samples for optimal performance (F1 score of 70.13), incurs a total annotation cost of \$192.74.

Sentence: With this company he was developing data-mining and database technology , more specific high-level ontologies for intelligence and automated natural language understanding .

Target Annotation: With this company he was developing <entity type="field">data-mining</entity> and <entity type="field">database</entity> technology , more specific high-level ontologies for intelligence and <entity type="task">automated natural</entity> <entity type="task">language understanding</entity> .

ICA: With this company he was developing <entity type="task">data-mining</entity> and <entity type="field">database technology</entity> , more specific <entity type="miscellaneous">high-level ontologies</entity> for <entity type="field">intelligence</entity> and automated <entity type="task">natural language understanding</entity> .

ICA w/ Self-consistency: With this company he was developing <entity type="field">data-mining</entity> and <entity type="field">database technology</entity> , more specific <entity type="miscellaneous">high-level ontologies</entity> for intelligence and automated <entity type="task">natural language understanding</entity> .

ICA w/ Self-correction: With this company he was developing <entity type="field">data-mining</entity> and <entity type="field">database technology</entity> , more specific <entity type="miscellaneous">high-level ontologies</entity> for intelligence and automated <entity type="task">natural language understanding</entity> .

ICA w/ EAR: With this company he was developing <entity type="field">data-mining</entity> and <entity type="field">database technology</entity> , more specific high-level ontologies for intelligence and automated <entity type="task">natural language understanding</entity> .

Figure 14: Case study from the AI dataset illustrating how the self-refinement methods correct annotation.

In contrast, our ICA method achieves a higher F1 score (76.12) using only 100 human-labeled samples combined with 1,000 LLM-annotated samples. This setup results in a significantly reduced total annotation cost of only \$46.98 (human annotation: \$42.85 for 100 samples; LLM annotation: \$0.413 per 100 samples, totaling \$4.13 for 1,000 samples).

This cost analysis highlights that the ICA framework substantially lowers annotation expenses while delivering superior performance. More importantly, by separating the costly LLM annotation from inference, our method directly addresses real-world deployment considerations regarding cost-effectiveness and scalability.

B.6 Class-wise performance

Table 5 reports per-entity-type F1 across 14 AI-domain categories. ICA w/ EAR attains the best score on 12/14 types; the two exceptions are field (where ICL-MS is slightly higher) and organisation (where ICA is marginally ahead). On average, ICA w/ EAR reaches 0.7964 F1—an absolute gain of +0.056 over ICL-MS (0.7403; +7.6% relative) and +0.034 over ICA (0.7625; +4.4% relative). The largest margins (vs. ICL-MS) occur on long-tail or ambiguity-prone categories—location (+0.25), metrics (+0.09), misc (+0.11), task (+0.08), person (+0.09), and product (+0.09)—indicating improved robustness beyond head entities. We attribute these gains to EAR’s targeted repairs of missing, spurious, and typing errors (prompts and settings in the

appendix). Overall, the improvements span both proper-noun types (e.g., person, organisation, country) and technical/semantic types (e.g., algorithm, metrics, task), supporting the generality of our approach rather than gains concentrated in a narrow subset of labels.

C AI Assistants In Writing

We utilized ChatGPT to identify writing issues and polish the manuscript for clarity and conciseness.

label	algorithm	conference	country	field	location	metrics	misc	organisation	person	product	programlang	researcher	task	university	Avg.
ICL-ZS	0.5675	0.7194	0.7863	0.6781	0.0286	0.0493	0.1596	0.6292	0.4000	0.6213	0.7480	0.8328	0.5660	0.5882	0.5779
ICL-MS	0.7739	0.8346	0.8892	0.8268	0.5600	0.7622	0.3387	0.8140	0.8108	0.6616	0.7286	0.9097	0.6661	0.8971	0.7403
ICA	0.7754	0.8468	0.8372	0.8033	0.7438	0.8183	0.3938	0.8531	0.8750	0.6935	0.8052	0.9325	0.7110	0.9032	0.7625
ICA w/ EAR	0.8150	0.8562	0.8905	0.8200	0.8099	0.8480	0.4444	0.8526	0.9038	0.7493	0.8235	0.9414	0.7467	0.9467	0.7964

Table 5: Per-class performance on the AI domain.