A Systematic Comparison of Large Language Models for Data Annotation in NER Tasks

Muhammad Uzair Ul Haq[⋆], **Davide Rigoni**[†], **Alessandro Sperduti**[†§‡•]
[⋆]Amajor SB S.p.A, Italy
[†]Department of Mathematics "Tullio Levi-Civita", University of Padova, Italy
[§]Human Inspired Technology Research Centre, University of Padova, Italy
[‡]Department of Information Engineering and Computer Science, University of Trento, Italy
[•]Fondazione Bruno Kessler, Italy
ulhaquzair@gmail.com, {davide.rigoni.1, alessandro.sperduti}@unipd.it

Abstract

High-quality annotated data is essential for training effective machine learning models, especially for fine-grained tasks like Named Entity Recognition (NER), where each token in a sentence must be tagged with a golden annotation. While Large Language Models (LLMs) show strong potential in automating data annotation, existing literature lacks extensive evaluations that systematically compare different models, embedding strategies, and context selection methods, particularly on complex, real-world datasets. This paper fills this gap by conducting a comprehensive study of LLMs for NER annotation across four diverse datasets. It benchmarks both proprietary and open-source LLMs at the 7B to 70B parameter scale, including a 32B reasoning-optimized model, and explores multiple context selection strategies. Two evaluations are performed: (i) the assessment of the practical utility of LLM-generated annotations by fine-tuning a RoBERTa model on LLM-generated annotations and measuring downstream performance; (ii) the assessment of only LLM-generated annotations using token-level metrics, like Precision, Recall, F_1, and agreement with human annotations (Cohen's κ). Empirical results, supported by statistical tests, highlight the importance of choosing suitable LLMs and embedding models and reveal key trade-offs between model scale and annotation quality. Challenging datasets like SKILLSPAN further expose the limitations of current LLM-based annotation pipelines, emphasizing the need for benchmarking on difficult, real-world tasks.

Keywords: Information Extraction, Data Annotation, NER Evaluation, RAG

1. Introduction

Data annotation plays a crucial role in training ML models. In NLP, data annotation typically involves annotating text data with relevant information, such as named entities, parts of speech, sentiment, intent, text classification, etc. The process of annotating data carries even greater significance for fine-grained NLP tasks, such as token classification, where each token in a sentence must be tagged with a gold label.

In specialized domains such as human resource management, medical (Liu et al., 2022), finance (Loukas et al., 2022), etc., organizations often possess large datasets that can be leveraged to enhance decision-making and operational efficiency through the use of LLM-based approaches (Urlana et al., 2024). Consequently, companies are increasingly adopting fine-tuning as a strategic approach, as it enables them to optimize model performance, reduce dependency on costly external APIs, and tailor solutions to meet industry-specific requirements, thereby ensuring data privacy and alignment with corporate objectives. However, for these organizations to fully harness the power of LLMs through fine-tuning, high-quality annotated datasets are needed, which is a labor-intensive and costly process, especially when applied to large corpora (Feng et al., 2021a). With the advent of advanced LLMs such as GPT (OpenAI, 2023), Qwen (Team, 2024b), DeepSeek (DeepSeek-AI, 2025), and Llama (Touvron et al., 2023) families, researchers and practitioners are increasingly leveraging these models to enhance the data annotation process (Tan et al., 2024), as they offer unparalleled capabilities for automation, scalability, and cost reduction (Wang et al., 2021).

Recent studies have demonstrated that LLMs (Wang et al., 2023; Naraki et al., 2024) can be effectively employed for data annotation in the NER task, achieving performance comparable to human-level in some cases. However, the authors argue, and the findings of the proposed study further support, that these conclusions are largely based on evaluations using relatively simple benchmark datasets, such as CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and WNUT-17 (Derczynski et al., 2017). Notably, between 2023 and 2025, the CoNLL-2003 dataset was used in 191 studies, while WNUT-17 appeared in 45 studies. In contrast, more complex datasets like SKILLSPAN (Zhang et al., 2022a) and GUM (Zeldes, 2017) have been used significantly less frequently, appearing in only 9 and 4 studies, respectively[1]. Moreover,

[1]Statistics from https://paperswithcode.com/dataset/ as of August 2, 2025.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 524–548
11–16 May 2026.

while standard performance metrics such as Precision, Recall, and F_1 score offer useful insights into the annotation quality of LLMs, they alone are insufficient for a comprehensive evaluation. These metrics primarily capture surface-level correctness but do not fully reflect the alignment of LLM-generated annotations with human judgment. Thus, a more rigorous analysis is needed to evaluate the effectiveness of LLMs in NER.

Motivated by the lack of a comprehensive study in the literature, this article systematically examines the performance of LLMs in data annotation for NER tasks. The main goal is to establish a robust and reproducible baseline to guide future research in this area. To create a strong baseline, it extends the evaluation beyond conventional metrics such as Precision, Recall, and F_1-score, adopting a systematic approach to assess the reliability of LLM-generated annotations. First, it conducts a practical downstream evaluation by fine-tuning a RoBERTa model on both human-annotated and LLM-generated data. This experiment simulates real-world scenarios where LLMs are employed for annotation, allowing the assessment of the practical implications of using such labels in model training. Then, to further validate the findings, it directly evaluates the quality of gold labels produced by LLMs by comparing them against human-annotated data. This involves measuring inter-annotator agreement as well as standard classification metrics, providing insights into the degree of alignment between LLM-generated and human annotations.

The main contributions of this study are summarized as follows: (i) it benchmarks a range of LLMs for NER annotation across four diverse datasets. The evaluation encompasses both proprietary and open-source models, spanning parameter scales from 7B to 70B, including a reasoning-optimized variant; (ii) it evaluates the practical utility of LLM-generated annotations by training a downstream model, RoBERTa, on these labels. This analysis investigates different embedding models and context selection strategies, namely zero-shot, in-context learning, and retrieval-augmented generation, supported by statistical significance testing; (iii) it further evaluates the LLM-generated labels using standard metrics, such as Precision, Recall, and F_1, complemented by Cohen's k to provide a comprehensive assessment of their alignment with human annotations; (iv) it reveals that more challenging datasets, such as SKILLSPAN, expose the limitations of LLM-generated annotations and context retrieval methods.

2. Related Work

In the recent past, there have been efforts by researchers to leverage the LLMs for data annotation (Tan et al., 2024). Wang et al. (2021) introduced the use of GPT-3 (Brown et al., 2020) for data annotation by evaluating its generated data against human-labeled data. To annotate each sentence with the model, the authors construct a prompt that includes several human-labeled examples along with the target sentence, and evaluate the performance in n-shot settings. Additionally, they report the performance of text classification and data generation tasks. Likewise, He et al. (2024) leveraged GPT-3.5-based models to annotate data and, in comparison to the previous approach, introduced the concept of Chain-of-Thought (CoT) (Wei et al., 2023) reasoning for this purpose. The idea is to simulate the human reasoning process to induce GPT-3.5 to motivate the annotated examples. They provide GPT-3.5 with the task description, specific examples, and the corresponding gold labels, then ask the model to explain whether and why the given label is appropriate for that example. This enables the model to explain its choice of a specific label for the target sentence. Then, the authors construct the few-shot CoT prompts using the explanations generated by the model for data annotation.

Following the work of Wang et al. (2021) and Wang et al. (2023), Naraki et al. (2024) also proposed an LLM-based annotation for the NER task. The authors used the LLMs to clean noise and inconsistencies in the NER dataset, and then they merged the cleaned NER dataset with the original dataset to generate a more robust and diverse set of annotations. It is worth mentioning that, in merging the annotations from LLM with human labels, preference is given to human-annotated examples compared to the LLM annotations. In addition, Bogdanov et al. (2024) used the LLMs to create a general dataset for NER tasks with a broad range of entity types. The authors presented a procedure that consists of annotating raw data with an LLM to train a task-specific foundation model for NER. Goel et al. (2023) use the same concept of data annotation using LLMs; however, they apply it to a medical domain for accelerating the annotation process along with human input. Lastly, (Dao et al., 2025) addressed the data scarcity problem through LLM-generated synthetic data.

These studies highlight a strong interest in using LLMs for dataset annotation. However, a systematic evaluation of complex NER datasets with several LLMs remains unexplored, which motivates the proposed study.

3. Methodology

3.1. Problem Formalization

The NER task can be defined as the problem of learning an approximation function $\widetilde{f}_\theta$ that closely matches the real function $f : \mathcal{S}_\mathcal{V} \times \mathcal{I} \rightarrow \mathcal{C}$, where $\mathcal{S}_\mathcal{V}$ is the set of all sentences formed from the vocabulary $\mathcal{V}$, $\mathcal{I}$ is the set of word positions, and $\mathcal{C}$ is the set of entity categories. The real function f given: (i) a sentence $S = (w_1, \ldots, w_z) \in \mathcal{S}_\mathcal{V}$ composed of z words, and (ii) an index $j \in \{1, \ldots, z\}$ identifying a specific occurrence of a word in S, returns the entity label $c \in \mathcal{C}$ of that word w_j.

This study divides a dataset of sentences into three groups: training, validation, and testing. The training set is then further divided into two splits: one for selecting context examples and another for annotation purposes. The goal is to select a subset of context examples that will guide an LLM in annotating the rest of the sentences in the training set. Formally, it is assumed to be available a dataset $\mathcal{D} = \{S_i\}_{i=1}^n$, where S_i represents the i-th sentence, with training, validation and test split given as $\mathcal{D}_{train}$, $\mathcal{D}_{valid}$ and $\mathcal{D}_{test}$. The dataset $\mathcal{D}_{train}$ is divided into two disjoint subsets: $\mathcal{X}$, referred to as the sample space, from which context examples are sampled, and $\mathcal{T}$, which will be annotated by the LLM. Formally, let $\mathcal{X} \subset \mathcal{D}_{train}$ be a subset of size x, where $x < n$, and $\mathcal{T} = \mathcal{D}_{train} \setminus \mathcal{X}$ be the remaining subset containing t sentences, where $t = n - x$. From $\mathcal{X}$, m examples are selected, where $m < x$, to form the context set $\mathcal{M}$. The LLM uses all the m examples in $\mathcal{M}$ as input context to annotate the t sentences in $\mathcal{T}$.

3.2. Approaches to Data Annotation

Crafting an effective prompt for LLMs is a crucial task, as different LLMs tend to respond differently to prompt variations (Errica et al., 2024). This work adopts the same approach to prompt design used in He et al. (2024) and Wang et al. (2023).

In the zero-shot setting (i.e., the baseline), LLMs receive only task descriptions and entity categories from the dataset. This setting evaluates the models' general capabilities without any explicit task-specific adaptation. It is important to note that because modern LLMs are pre-trained on massive, partially undisclosed corpora, it is possible that some evaluation instances (or closely related content) were seen during pre-training. This potential data contamination is difficult to verify, especially for proprietary models. For this reason, we treat zero-shot performance as a reference point and focus on the relative improvements obtained by more advanced techniques such as In-Context Learning (ICL) and RAG. In In-Context Learning, the prompts given to LLMs are enhanced by including not only a task description and entity categories but also $\mathcal{M}$ contextual examples sampled from $\mathcal{X}$. The sampling can be approached in two ways: either through manual cherry-picking or by random sampling, although cherry-picking contradicts the rationale of this article. Thus, random sampling is used. To address the limitations of the aforementioned approaches, this study also considers a RAG-based method to automatically select relevant contextual examples via cosine similarity.

3.3. Evaluation Settings

To comprehensively evaluate the quality of LLM-generated annotations, this study adopts a multi-step evaluation that not only captures the practical utility in real-world downstream tasks but also provides a deeper insight into errors made by LLMs.

Downstream NER Scenario To assess the practical utility of LLM-generated annotations, this work conducts a downstream evaluation using an encoder-based model. Specifically, this study adopts the RoBERTa (Liu et al., 2019) model due to its widespread use in NER tasks (Zhou and Li, 2022; Zhang et al., 2022a; Ul Haq et al., 2024; Haq et al., 2024). The evaluation pipeline is as follows: for a given dataset $\mathcal{D}_{train}$, a subset $\mathcal{T} \subset \mathcal{D}_{train}$ is selected, which is annotated automatically using an LLM, following the strategies described in Section 3.2. The resulting annotated subset $\hat{\mathcal{T}}$ is then used to fine-tune RoBERTa. The Model Selection is based on the validation set $\mathcal{D}_{valid}$, and the final evaluation is conducted on the held-out test set $\mathcal{D}_{test}$. To ensure robustness, this study reports the average performance across five random seeds. Conceptually, this evaluation provides insights into how well LLM-generated annotations can support real model training.

LLM Post-processing Errors LLM-generated annotations must conform to the correct output format, ensuring precise token-level alignment with the gold labels. Even minor discrepancies, such as incorrect span boundaries or improperly placed tags, can substantially distort evaluation metrics and compromise downstream applications. Consequently, this evaluation rigorously examines the structural reliability of various LLMs by assessing their adherence to the annotation format and compliance with task-specific constraints.

LLM-generated Annotations This evaluation assesses the quality of the generated annotations for each LLM using direct token-level metrics:

Dataset	Sentences			Average Entity Length
	Train	Validation	Test	
CoNLL-2003	14041	3250	3453	1.60
WNUT-2017	3394	1008	1287	1.73
GUM	1435	615	805	3.15
SKILLSPAN	3074	1396	1522	4.72

Table 1: Dataset statistics, with Average Entity Length as the mean number of tokens per entity.

Precision, Recall, F_1, and Cohen's k scores. These are computed by comparing LLM-generated annotations with the human-annotated gold labels across multiple datasets. The Precision, Recall, and F_1 metrics offer a straightforward measure of how well the LLM captures relevant named entities, although they provide only a partial picture. For this reason, Cohen's k (Cohen, 1960) metric is also considered, as it is a statistical measure used to evaluate the level of agreement between two raters or observers on categorical classifications, accounting for the agreement occurring by chance. This metric highlights how well LLMs align with gold labels, offering deeper insight than token-level metrics alone.

4. Experimental Assessment

4.1. Datasets

To evaluate the performance of the proposed methodology and assess the capabilities of LLMs, four datasets are considered, with their statistics summarized in Table 1. Each dataset presents unique challenges for LLMs in performing NER tasks, allowing this study to comprehensively analyze the ability of LLMs to handle diverse entity types, from well-structured entities to complex, ambiguous, and domain-specific annotations. The CoNLL-2003 and WNUT-2017 datasets primarily feature single-token entities, with average lengths of 1.60 and 1.73 tokens, respectively. In contrast, GUM has more complex entities, averaging 3.15 tokens. SKILLSPAN is the most complex, with entity lengths averaging 4.72 tokens, indicating the need for advanced modeling techniques.

4.2. LLMs Under Study

In the empirical assessment of the datasets annotated by LLMs, the zero-shot data annotation approach is chosen as the baseline since it provides no context about the task to the LLM, thus allowing the evaluation of the LLM's general knowledge of the task. In addition, the ICL and RAG-based approaches, described in Section 3.2, are also considered. The experiments were conducted using three different numbers of context examples: (i) 25, (ii) 50, and (iii) 75, while fixing the subset size to $\mathcal{X} = 30\%$ of the training set $\mathcal{D}_{\text{train}}$. To further investigate the impact of varying $\mathcal{X}$, an ablation study was performed (Section 5) by considering smaller sample sizes of 10% and 20%.

This study considers six LLMs: (i) `gpt-4o-mini-2024-07-18`; (ii) `Qwen2.5-72B-Instruct`; (iii) `Llama3.1-70B-Instruct`; (iv) `DeepSeek-R1-Distill-Qwen-32B`; (v) `Qwen2.5-7B-Instruct`; and (vi) `Llama3.1-8B-Instruct`. Additionally, two embedding models are considered: *(i)* `text-embedding-3-large`[2]; and *(ii)* `sentence transformer all-MiniLM-L6-v2` (Reimers and Gurevych, 2019). Throughout the remainder of the paper, `text-embedding-3-large` will be referred to as OpenAI, and `sentence transformer all-MiniLM-L6-v2` will be referred to as ST. In the rest of the article, models are referred to by their base names, e.g., `Qwen2.5-72B` for `Qwen2.5-72B-Instruct`, and so on.

4.3. Implementation Details

To perform experiments for data annotation with `gpt-4o-mini`, the model is accessed via the API service provided by OpenAI. To ensure reproducible results, the temperature is set to 0, and a seed value of 42 is used. Furthermore, the system fingerprint `fp_1bb46167f9` is reported as noted during API access. For data annotation generation using Qwen (Team, 2024b) and Llama (Touvron et al., 2023) based models, the HuggingFace (Hugging Face, 2023) implementation is utilized. The experiments with billion-scale models are conducted on an $A100$ GPU with a seed value of 42. The instruction fine-tuned variants of the open-source models are employed in the proposed study. All models are used exclusively for inference and are uniformly quantized to 4-bit precision to ensure consistency across the open-source implementations. As discussed in Pecher et al. (2025) and Yazan et al. (2024), quantization has a negligible impact on performance for relatively simple tasks such as information extraction. Regarding the proprietary model, `gpt-4o-mini`, the quantization scheme is not publicly disclosed. All experiments to fine-tune the NER task are performed with the RoBERTa model available via Hugging Face and conducted in a Python environment on an RTX $A5000$ GPU. Each experiment is executed with early stopping with a patience value of 3 and a cut-off of 0.3. The batch size is found to be 32; the initial learning rate is set to 2×10^{-5} with AdamW optimizer (Kingma and Ba, 2014), and weight decay of 0.2. The different runs are

[2] https://platform.openai.com/docs/guides/embeddings

performed using the following five seed values: $[23112, 13215, 6465, 42, 5634]$. Moreover, the statistical significance tests are performed with the help of the scikit-posthocs (Terpilowski, 2019) library available in Python. The code is publicly available online[3].

4.4. Downstream NER Results

Figure 1 provides an overview of the experimental results conducted in the proposed study. The heatmaps illustrate the F_1 scores on the test sets across different datasets, comparing the performance of various models and methods evaluated in this study. The CoNLL-2003 dataset, which contains named entities like persons, organizations, and locations, is relatively well-structured, making it easier for LLMs to generate high-quality annotations. The `gpt-4o-mini` model with OpenAI embeddings emerges as the top performer, just 2.7% below human-level annotation, when considering 75 context examples. Among the 70B models, `Qwen2.5-72B` with OpenAI embeddings performs comparably to `gpt-4o-mini`, while `llama3.1-70B` with ST embeddings lags slightly behind. Interestingly, by harnessing its reasoning capabilities, `DeepSeek-R1-32B` performs comparably to 70B scale models. At the 7B scale, `Qwen2.5-7B` with ST embeddings significantly outperforms its counterpart with 70B when using 25 and 50 context examples. This suggests that smaller models can still perform competitively when paired with appropriate embedding methods. Interestingly, the heatmap reveals that context size plays a crucial role: `gpt-4o-mini` and `Qwen2.5-70B` benefit from the context sizes of 75 examples, while `llama3.1-70B` performs best at a lower context size. This suggests that models reach context saturation at different points, where additional examples may not always improve performance linearly.

The WNUT-17 dataset, which focuses on low-frequency and emerging entities, presents a significant challenge due to limited training samples for each entity. However, `DeepSeek-R1-32B` with OpenAI embeddings achieves the highest score, outperforming `gpt-4o-mini` and `Qwen2.5-70B`. The `llama3.1-70B` model exhibits inconsistent performance, scoring only 51.18 with ICL at 75 context examples, suggesting that it struggles to generalize well to rare entities. At the 7B scale, `Qwen2.5-7B` with ST embeddings significantly outperform `Llama3.1-8B`. This highlights that ST embeddings provide a crucial advantage for smaller models. Compared to human-level annotation, which achieves an F_1 score of 54.93, the best-performing LLM, which adopts RAG, reduces the gap to just 1.21%, which is the smallest performance gap between human and LLM annotation across all datasets used in the experiments. This suggests that RAG-based annotation is highly effective in adapting to rare entity recognition, particularly when combined with larger models and appropriate embeddings.

The GUM dataset presents a unique challenge due to its diverse entity types, requiring models to generalize across various linguistic structures. `DeepSeek-R1-32B` with OpenAI embeddings surpasses the rest of the models. At the 7B scale, `Qwen2.5-7B` with OpenAI embeddings outperforms `Llama3.1-8B`. However, both models perform noticeably worse than their larger counterparts, indicating that smaller models have difficulty with datasets containing diverse entities. The 3% gap between the top-performing LLM and human annotation indicates that GUM is still challenging for LLMs. Moreover, the heatmap shows that model performance varies significantly with context size and embedding choice.

The SKILLSPAN dataset is the most difficult, as it requires understanding nuanced skill mentions across various job contexts. `gpt-4o-mini` with OpenAI embeddings performs the best with 75 context examples, but this is still far from human-level annotation. At the 70B scale, `Qwen2.5-70B` with either ST and OpenAI embeddings outperforms `llama3.1-70B`, while being comparable in the other settings. Among 7B models, `Qwen2.5-7B` with OpenAI embeddings significantly surpasses `Llama3.1-8B`. This suggests that embedding choice plays a crucial role in skill extraction tasks. Notably, the gap between human annotation and the best-performing LLM is much larger in this dataset compared to others, indicating that LLMs struggle with skill-based entity recognition. This could be due to the complexity of contextual skill interpretation, requiring deeper domain knowledge and better understanding capabilities.

To robustly evaluate the significance of the findings presented in this section, statistical hypothesis testing was performed on all reported results. Initially, the Friedman test was used to rank model accuracies across dataset runs and assess the null hypothesis of model equivalence. This was followed by Conover's post-hoc test (Pereira et al., 2015), with the outcomes illustrated in Figure 2. This statistical test highlights a crucial aspect: a trade-off when addressing the NER task. Indeed, larger models, such as those with 70B parameters, may not necessarily offer better performance than smaller models like `Llama3.1-8B-ST` or `Qwen2.5-7B`. This implies that the extra

[3] https://github.com/UzairUlhaq/LLM_Annotator

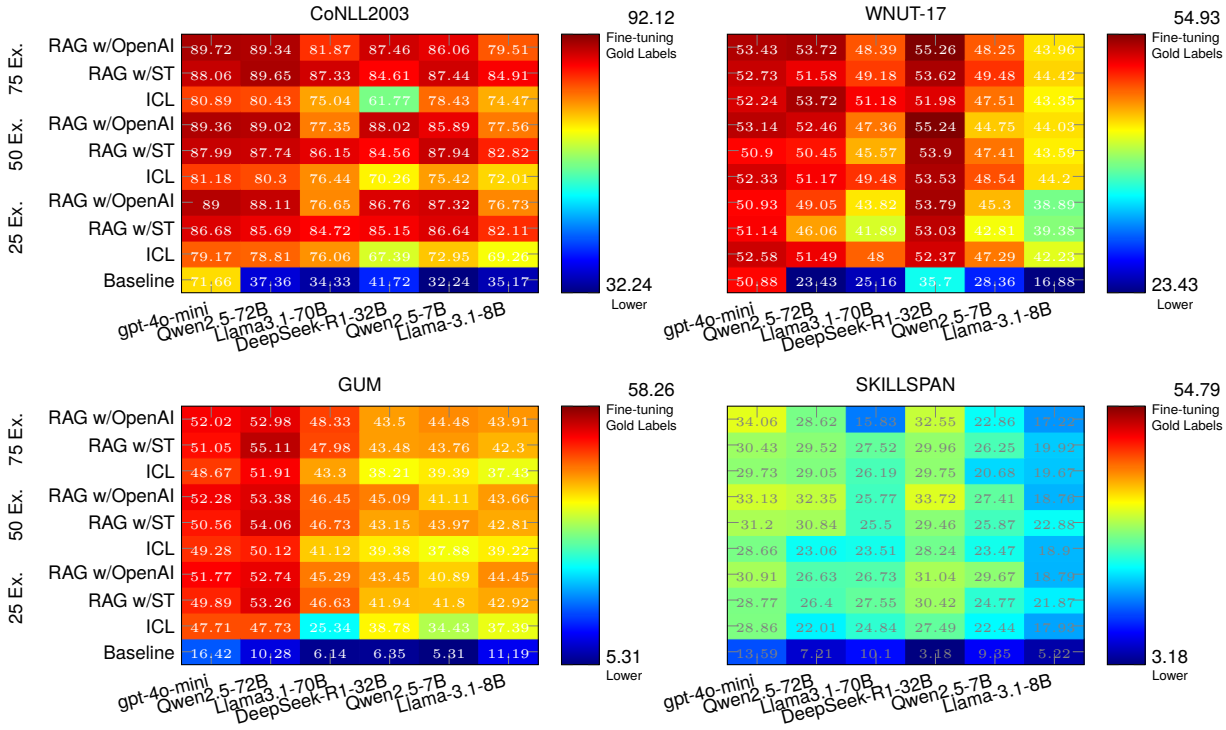

Figure 1: Heatmaps of the F_1 scores across four datasets. Blue denotes the lower F_1 score, whereas red denotes the higher F_1 score achieved by fine-tuning the RoBERTa model directly on the gold labels.

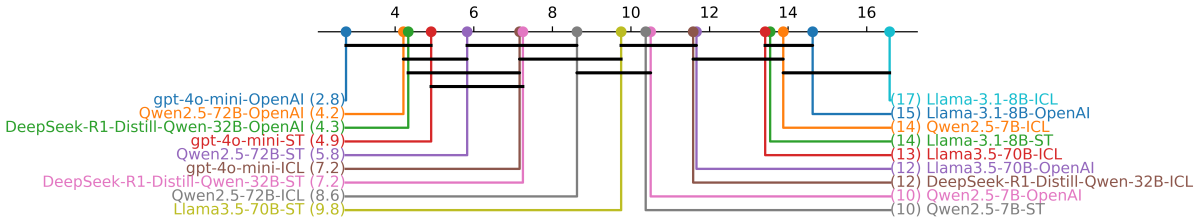

Figure 2: Critical difference diagram of average score ranks. Models with lower ranks perform better than those with higher ranks, while models linked by a horizontal line have no statistically significant difference. Labels are reported in the format "`LLM – Size, if available – Embedding Model`".

computational cost of larger models may not always be justified, particularly if smaller models can deliver statistically similar results.

Overall, the key insights are: (i) RAG-based approaches improve annotation quality, with `gpt-4o-mini` and OpenAI embeddings achieving the best results. In contrast, ICL struggles in datasets with sparse or ambiguous entities; (ii) while all models perform well on CoNLL-2003, performance declines as entity structures become more complex, such as in SKILLSPAN. This highlights the urgent need to increase the use of challenging datasets in evaluations; (iii) the choice of embeddings for the retrieval of context for LLMs plays a crucial role in annotation quality in retrieval-based methods; (iv) larger models do not always yield better performances. `Qwen2.5-7B` slightly outperforms `Llama3.1-8B` and performs comparably to `llama3.1-70B` when equipped with suitable embeddings. Furthermore, the reasoning model `DeepSeek-R1-32B-OpenAI` performs on par with significantly larger variants such as `Qwen2.5-72B`. This suggests that model architecture and training data significantly influence annotation quality, beyond parameter count.

4.5. LLM Post-processing Results

NER is a highly sensitive task where maintaining token-level alignment between model-generated annotations and gold labels is essential. Even minor misalignments, such as incorrect span boundaries or misplaced tags, can lead to significantly distorted evaluation metrics and hinder downstream applications. This section evaluates the structural reliability of different LLMs by analyzing their adherence to the

Model	Dataset			
	CoNLL2003	WNUT-17	GUM	SKILLSPAN
gpt-4o-mini	0.00	0.00	0.00	0.00
Qwen2.5-72B	0.33	0.15	1.45	0.74
Llama3.1-70B	0.89	1.97	1.61	2.15
DeepSeek-R1-32B	7.71	8.56	9.60	10.90
Qwen2.5-7B	3.02	0.78	1.33	0.56
Llama3.1-8B	7.02	9.75	10.16	4.27

Table 2: Percentage of examples discarded due to structural inconsistencies in post-processing.

annotation format and task-specific constraints. Specifically, the focus is on LLMs using RAG approaches paired with OpenAI embeddings, given the superior performance that emerged from Figure 1. To enable fair comparison, the structured output capability of LLMs is used to ensure a consistent output schema. However, as Table 2 shows, adherence to these instructions varies considerably across models. The findings indicate that larger models generally exhibit better formatting consistency. Notably, `gpt-4o-mini` demonstrates perfect structural adherence across all datasets with a 0% error rate. Among the open-source models, the `Qwen` family consistently shows low processing error rates and high fidelity to the output format. In contrast, the `Llama` models frequently violate annotation conventions, resulting in higher structural error rates, particularly on datasets that contain more diverse and noisy entity types. The `DeepSeek-R1-32B` model, despite its large size, is optimized for reasoning rather than structured generation. As a result, it does not natively support format-constrained output, requiring extensive manual cleaning.

4.6. LLM-generated Annotations Results

This section provides a detailed analysis of the most frequent errors made by LLMs compared to human-annotated gold labels. The focus is on LLMs using RAG approaches paired with OpenAI embeddings, given the superior performance that emerged from Figure 1. The aim is to understand not only *what* types of mistakes are common, but also *why* these errors occur. The analysis focuses on fine-grained, token-level insights across datasets, highlighting consistent failure modes and their implications.

Table 3 compares the performance of various LLMs against human-annotated gold labels, considering Precision, Recall, F_1, and Cohen's k metrics. A consistent pattern is that models tend to overpredict entity spans, leading to high Recall values, but negatively impacting Precision and F_1 scores. CoNLL-2003 shows the highest overall scores, with `Qwen2.5-72B` achieving the best F_1 and k values. The SKILLSPAN dataset is the most challenging for all models because of its ambiguous `SKILL` category. Despite this, `gpt-4o-mini` performs the best, though overall scores are still low.

The results underscore several trends. Cohen's k values closely mirror F_1 trends, reinforcing the idea that annotation quality depends not only on correctness but also on reliability. High k scores — such as those for CoNLL-2003 — correlate with clearer task definitions and stronger model performance, while lower k scores in datasets, like SKILLSPAN, reflect both the complexity of the task and the divergence between LLM and gold annotations. Together, these metrics provide a holistic view of model behavior, capturing both annotation accuracy and consistency.

Further analysis of token-level errors is presented in Figure 3, which illustrates the most frequent misclassification patterns across models and datasets. The error percentage is calculated over each entity. A prominent pattern is the frequent mislabeling of non-entity tokens `O` as part of an entity. For instance, in the CoNLL-2003 dataset, a pattern is observed regarding the annotation of non-entity tokens as part of a `MISC` entity — specifically transitions from `O` to `B-MISC` and `O` to `I-MISC`. Among all evaluated models, `gpt-4o-mini` shows the lowest error rate for this category (7.4%), while `Llama3.1-8B` exhibits the highest (15.8%). This highlights the inherent ambiguity of the `MISC` label, which often includes loosely defined categories such as nationalities, locations, or organizations, making it more challenging to disambiguate than more concrete classes like `PER` or `LOC`. Likewise, in the SKILLSPAN dataset, it is visible that the highest error percentages occur due to `O` tokens being misclassified as `I-Skill` and `B-SKILL`. These results indicate that, despite clear instructions and contextual examples, models continue to struggle with the consistent identification of ambiguous entity types. The WNUT-17 and GUM also show a similar trend.

When considering the adherence to structured output formats, as discussed in Section 4.5, another intriguing observation emerges: `DeepSeek-R1-32B` demonstrates relatively strong performance despite exhibiting high structural inconsistencies. This suggests that strong general reasoning and task adaptation capabilities can partially compensate for output formatting issues. Most reasoning-focused models currently lack native support for structured output[4] generation; a critical limitation for tasks like NER that require precise and consistent formatting. Pairing these models with mechanisms enforcing structured outputs could significantly enhance

[4] https://openai.com/index/introducing-structured-outputs-in-the-api

Model	CoNLL2003				WNUT-17				GUM				SKILLSPAN			
	Prec.	Recall	F_1	k	Prec.	Recall	F_1	k	Prec.	Recall	F_1	k	Prec.	Recall	F_1	k
gpt-4o-mini	**86.70**	89.42	88.04	90.99	**42.68**	78.45	**55.29**	**61.09**	**50.70**	60.05	**54.98**	**59.24**	21.42	48.41	**29.70**	**42.85**
Qwen2.5-72B	84.43	94.60	**89.23**	**91.63**	34.37	**79.98**	48.08	56.32	47.71	**60.92**	53.51	57.72	20.82	48.23	29.09	41.93
Llama3.1-70B	68.98	95.45	80.08	82.85	22.49	79.29	35.04	41.96	42.73	60.31	50.02	53.44	**20.93**	48.51	29.24	37.69
DeepSeek-R1-32B	78.53	89.82	83.80	87.34	40.29	75.92	52.65	59.18	37.51	46.11	41.37	45.86	19.79	46.25	27.72	41.39
Qwen2.5-7B	62.11	90.12	73.53	77.55	20.82	73.54	32.45	39.37	36.57	47.29	41.25	46.42	8.36	**54.65**	14.49	27.29
Llama3.1-8B	68.28	**95.99**	79.79	80.67	17.61	77.16	28.67	31.94	39.05	52.51	44.79	48.49	7.97	13.45	8.80	25.29

Table 3: Comparison of LLM-generated annotations with gold labels. Bold indicates the highest values.

their performance, enabling them to leverage both semantic reasoning and formal task adherence.

Overall, three key insights can be summarized. First, current models rarely confuse entity *types* — e.g., mistaking `PER` for `LOC` — rather, they frequently misidentify entity *boundaries* or generate spurious entities. Second, the overprediction trend underscores the importance of improving span-level reasoning and boundary detection. Lastly, despite the higher rate of post-processing errors, the reasoning model's performance suggests it may compensate for its lack of structured output through its reasoning capabilities.

5. Ablation Study

Tables 4 examine the influence of sample space $\mathcal{X}$ and context $\mathcal{M}$ sizes on entity recognition performance using the best-performing model, `gpt-4o-mini`, on the SKILLSPAN dataset. Increasing the context size from 25 to 75 generally improves the F_1 score, though gains diminish beyond 50 examples. RAG consistently outperforms ICL in Recall and F_1 score, demonstrating its effectiveness in leveraging external knowledge, while ICL achieves higher Precision but lower Recall, suggesting a more conservative prediction approach. At a 10% sample space, ICL delivers competitive results, but as it increases to 20%, RAG maintains a clear advantage, achieving the highest F_1 score with a context size of 75. Notably, for smaller dataset splits, RAG exhibits greater variability, similar to ICL, suggesting that when fewer examples are available, their performances converge.

6. Conclusions and Future Work

This study evaluates LLMs for automated NER data annotation through two main analyses. First, it measures the practical value of LLM-generated annotations by fine-tuning a RoBERTa model and assessing downstream performance across four datasets of varying complexity. Six LLMs of different parameter scales are benchmarked, alongside comparisons of embedding models and context selection strategies. Second, an intrinsic evaluation examines token-level metrics and agreement with human annotations, revealing that LLMs often over-predict entity spans—favoring Recall over Precision—and frequently make boundary and spurious entity errors. Statistical results indicate that well-tuned 7B models can match the performance of larger ones, suggesting potential for efficient NER annotation pipelines.

Future work will extend the analysis to additional LLM families, NLP tasks, and specialized domains such as medical NER.

Sample Space	Context Size	Precision	Recall	F_1 Score
		RAG		
10%	25	$21.83_{\pm 1.22}$	$56.94_{\pm 1.17}$	$31.53_{\pm 1.18}$
	50	$22.44_{\pm 1.32}$	$56.46_{\pm 2.46}$	$32.07_{\pm 1.13}$
	75	$22.82_{\pm 0.58}$	$55.82_{\pm 1.40}$	$\mathbf{32.34}_{\pm 0.41}$
20%	25	$20.26_{\pm 1.55}$	$54.46_{\pm 3.71}$	$29.45_{\pm 1.29}$
	50	$21.00_{\pm 0.81}$	$57.40_{\pm 1.57}$	$30.74_{\pm 0.86}$
	75	$22.69_{\pm 0.46}$	$56.31_{\pm 2.06}$	$\mathbf{32.39}_{\pm 0.60}$
		ICL		
10%	25	$22.57_{\pm 1.49}$	$48.72_{\pm 3.95}$	$30.74_{\pm 0.87}$
	50	$23.62_{\pm 0.85}$	$50.73_{\pm 1.33}$	$\mathbf{32.21}_{\pm 0.67}$
	75	$23.12_{\pm 1.16}$	$51.09_{\pm 4.19}$	$31.76_{\pm 0.89}$
20%	25	$19.35_{\pm 1.57}$	$45.83_{\pm 4.74}$	$27.05_{\pm 0.66}$
	50	$22.02_{\pm 1.56}$	$51.17_{\pm 1.15}$	$30.76_{\pm 1.39}$
	75	$22.89_{\pm 1.11}$	$49.78_{\pm 2.89}$	$\mathbf{31.32}_{\pm 0.99}$

Table 4: Comparison of RAG and ICL across various sample spaces and context sizes on SKILLSPAN dataset.

Limitations

This study evaluates LLMs for data annotation tasks and introduces a RAG-based approach with different embedding models to enhance performance on NER datasets. However, the proposed work has several limitations that highlight areas for future research. First, the experiments focus solely on NER tasks. While this provides a solid foundation for evaluation, extending the analysis to other NLP tasks, such as text classification or question answering, would offer a more comprehensive understanding of the proposed methodology's applicability and generalization. Second, for the proof of concept, this work employs a naive RAG approach for context selection. Indeed, the use of more sophisticated techniques, such as adaptive retrieval

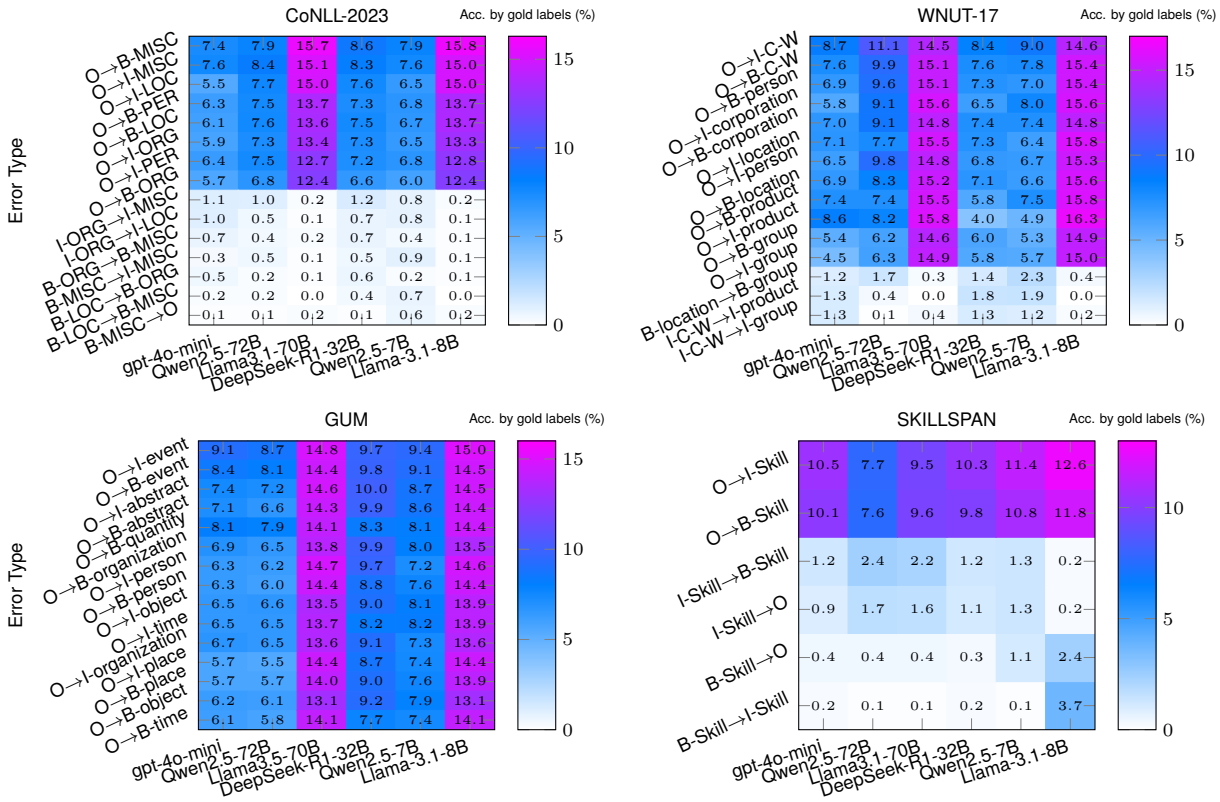

Figure 3: Most frequent named-entity prediction errors across four datasets. Each heatmap shows the percentage of each error type (gold label → predicted label) by model.

strategies, re-ranking mechanisms, or hybrid approaches combining dense and sparse retrieval, may further optimize performance. Third, this study does not explicitly examine the biases introduced by LLMs in the data annotation process. Given the growing concerns about fairness and model biases, a deeper investigation into how LLMs influence annotation patterns, especially in underrepresented datasets, could provide valuable insights.

Acknowledgements

The authors acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Concession Decree No. 1555 of October 11, 2022, CUP C63C22000770006. Moreover, the authors acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA.

7. References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Wahiba Karra Ben Abdessalem and Soumaya Amdouni. 2011. E-recruiting support system based on text mining methods. *International Journal of Knowledge and Learning*, 7(3-4):220–232.

Christian Bizer, Ralf Heese, Malgorzata Mochól, Radoslaw Oldakowski, Robert Tolksdorf, and Rainer Eckstein. 2005. The impact of semantic web technologies on job recruitment processes. In *Wirtschaftsinformatik*.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

William Jay Conover. 1999. *Practical Nonparametric Statistics*, volume 350. John Wiley & Sons.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Ryan Cotterell and Kevin Duh. 2017. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 91–96, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. 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.

An Dao, Hiroki Teranishi, Yuji Matsumoto, Florian Boudin, and Akiko Aizawa. 2025. Overcoming data scarcity in named entity recognition: Synthetic data generation with large language models. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, pages 328–340, Viena, Austria. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2022. Container: Few-shot named entity recognition via contrastive learning. In *ACL*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

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. *ArXiv*, abs/1810.04805.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, 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.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Federico Errica, Giuseppe Siracusano, Davide Sanvito, and Roberto Bifulco. 2024. What did i do wrong? quantifying llms' sensitivity and consistency to prompt engineering.

S. Fareri, Nicola Melluso, Filippo Chiarello, and Gualtiero Fantoni. 2021. Skillner: Mining and mapping soft skills from any text. *Expert Syst. Appl.*, 184:115544.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021a. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021b. A survey of data augmentation approaches for nlp. *ArXiv*, abs/2105.03075.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, Jean Steiner, Itay Laish, and Amir Feder. 2023. Llms accelerate annotation for medical information extraction.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Muhammad Uzair Ul Haq, Davide Rigoni, and Alessandro Sperduti. 2024. Prompt-based data augmentation using contrastive learning under scarcity of annotated data. In *ECAI 2024 - 27th European Conference on Artificial Intelligence*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 2717–2724. IOS Press.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hugging Face. 2023. Transformers APIs. `https://huggingface.co/docs/transformers/index`. Accessed: 2023-01-21.

Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. 2020. Pretraining with contrastive sentence objectives improves discourse performance of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4859–4870, Online. Association for Computational Linguistics.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713.

Janez Demsar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.*, 7:1–30.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh

Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Peng Jiang and Srikanth Saripalli. 2022. Contrastive learning of features between images and lidar. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, page 411–417. IEEE Press.

Mert Karabacak and Konstantinos Margetis. 2023. Embracing large language models for medical applications: Opportunities and challenges. *Cureus*, 15(5):e39305.

Imane Khaouja, Ismail Kassou, and Mounir Ghogho. 2021. A survey on skill identification from online job ads. *IEEE Access*, 9:118134–118153.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Amirhossein Layegh, Amir H. Payberah, Ahmet Soylu, Dumitru Roman, and Mihhail Matskin. 2023. Contrastner: Contrastive-based prompt tuning for few-shot ner. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 241–249.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Jin Li, Bingshi Li, Qirong Zhang, Xinlong Chen, Xinyang Huang, Longkun Guo, and Yang-Geng Fu. 2023. Graph contrastive representation learning with nbsp;input-aware and nbsp;cluster-aware regularization. In *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part II*, page 666–682, Berlin, Heidelberg. Springer-Verlag.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024. A simple but effective approach to improve structured language model output for information extraction.

Ning Liu, Qian Hu, Huayun Xu, Xing Xu, and Mengxin Chen. 2022. Med-bert: A pretraining framework for medical records named entity recognition. *IEEE Transactions on Industrial Informatics*, 18(8):5600–5608.

Pei Liu, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A survey of text data augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*, pages 191–195.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yu Liu, Duantengchuan Li, Kaili Wang, Zhuoran Xiong, Fobo Shi, Jian Wang, Bing Li, and

Bo Hang. 2024. Are llms good at structured outputs? a benchmark for evaluating structured output capabilities in llms. *Information Processing & Management*, 61(5):103809.

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Paliouras George. 2022. Finer: Financial numeric entity recognition for xbrl tagging. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*. Association for Computational Linguistics.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2024. An empirical study of catastrophic forgetting in large language models during continual fine-tuning.

Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online. Association for Computational Linguistics.

Lamiaa Mostafa and Sara Beshir. 2021. Job candidate rank approach using machine learning techniques. In *AMLTA*.

Yuji Naraki, Ryosuke Yamaki, Yoshikazu Ikeda, Takafumi Horie, and Hiroki Naganuma. 2024. Augmenting ner datasets with llms: Towards automated and refined annotation.

OpenAI. 2023. Gpt-4 technical report.

Branislav Pecher, Ivan Srba, and Maria Bielikova. 2025. Comparing specialised small and general large language models on text classification: 100 labelled samples to achieve break-Even performance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 165–184, Suzhou, China. Association for Computational Linguistics.

D. Pereira, Anabela Afonso, and Fátima Medeiros. 2015. Overview of friedman's test and post-hoc analysis. *Communications in Statistics - Simulation and Computation*, 44:2636–2653.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Davide Rigoni. 2023. Understanding multimedia content with prior knowledge.

Davide Rigoni, Luca Parolari, Luciano Serafini, Alessandro Sperduti, and Lamberto Ballan. 2023. Weakly-supervised visual-textual grounding with semantic prior refinement. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*, page 229. BMVA Press.

Davide Rigoni, Luciano Serafini, and Alessandro Sperduti. 2022. A better loss for visual-textual grounding. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 49–57.

Luiza Sayfullina, Eric Malmi, and Juho Kannala. 2018. Learning representations for soft skill matching. In *International Joint Conference on the Analysis of Images, Social Networks and Texts*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Oleksii Shatalov and Nataliya Ryabova. 2021. Named entity recognition problem for long entities in english texts. In *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, volume 1, pages 76–79.

Stefan Strohmeier. 2022. *Handbook of Research on Artificial Intelligence in Human Resource Management*. Edward Elgar Publishing.

Zhen Tan, Dawei Li, Alimohammad Beigi, Song Wang, Ruocheng Guo, Amrita Bhattacharjee, Bohan Jiang, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation: A survey. *ArXiv*, abs/2402.13446.

Gemini Team. 2024a. Gemini: A family of highly capable multimodal models.

Qwen Team. 2024b. Qwen2.5: A party of foundation models.

Maksim Terpilowski. 2019. scikit-posthocs: Pairwise multiple comparison tests in python. *The Journal of Open Source Software*, 4(36):1169.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Muhammad Uzair Ul Haq, Paolo Frazzetto, Alessandro Sperduti, and Giovanni Da San Martino. 2024. Improving soft skill extraction via data augmentation and embedding manipulation. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, SAC '24, page 987–996, New York, NY, USA. Association for Computing Machinery.

Ashok Urlana, Charaka Vinayak Kumar, Ajeet Kumar Singh, Bala Mallikarjunarao Garlapati, Srinivasa Rao Chalamala, and Rahul Mishra. 2024. Llms with industrial lens: Deciphering the challenges and prospects – a survey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jiuniu Wang, Wenjia Xu, Xingyu Fu, Guangluan Xu, and Yirong Wu. 2020. Astral: Adversarial trained lstm-cnn for named entity recognition. *Knowledge-Based Systems*, 197:105842.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Ivo Wings, Rohan Nanda, and Kolawole John Adebayo. 2021. A context-aware approach for extracting hard and soft skills. *Procedia Computer Science*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey.

Dongjie Yang, Zhuosheng Zhang, and Hai Zhao. 2023. Learning better masking for better language model pre-training.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, Yu-Yang Liu, and Li Yuan. 2024. Llm

lies: Hallucinations are not bugs, but features as adversarial examples.

Mert Yazan, Suzan Verberne, and Frederik Situmeang. 2024. The impact of quantization on retrieval-augmented generation: An analysis of small llms.

Xing Yi, James Allan, and W. Bruce Croft. 2007. Matching resumes and jobs based on relevance models. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Kun Yu, Gang Guan, and M. Zhou. 2005. Resume information extraction with cascaded hybrid model.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Mike Zhang, Kristian Nørgaard Jensen, Sif Dam Sonniks, and Barbara Plank. 2022a. Skillspan: Hard and soft skill extraction from english job postings. In *North American Chapter of the Association for Computational Linguistics*.

Mike Zhang, Kristian Nørgaard Jensen, Rob van der Goot, and Barbara Plank. 2022b. Skill extraction from job postings using weak supervision. *ArXiv*, abs/2209.08071.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022c. Automatic chain of thought prompting in large language models.

Ran Zhou and Xin and Li. 2022. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.

A Datasets Statistics

Table 1 of the main article highlights the complexity of entity mentions across different datasets, as reflected in their average entity length. CoNLL-2003 and WNUT-2017 contain relatively short entities, with average lengths of 1.60 and 1.73 tokens, respectively, indicating that most entities are single-token mentions. In contrast, GUM exhibits greater complexity, with an average entity length of 3.15 tokens, suggesting the presence of multi-token entities. SKILLSPAN is the most complex dataset, with an average entity length of 4.72 tokens, implying more intricate entity structures that require advanced modeling techniques for accurate recognition. Follows the entity information for each dataset.

CoNLL-2003 The CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) dataset consists of general entity types: *(i)* PERSON ; *(ii)* ORGANIZATION; *(iii)* LOCATION; and *(iv)* MISCELLANEOUS. Entities in this dataset typically follow structured patterns, making them relatively easier for LLMs to identify and classify.

WNUT-17 The WNUT-17 (Derczynski et al., 2017) dataset contains six categories of rare entities: *(i)* PERSON; *(ii)* CORPORATION; *(iii)* LOCATION; *(iv)* CREATIVE_WORK; *(v)* GROUP; and *(vi)* PRODUCT. This dataset is particularly challenging due to its noisy text, sparse entity occurrences, and limited labeled examples per category.

GUM The GUM (Zeldes, 2017) dataset is a richly annotated corpus designed for multiple NLP tasks, including NER. The dataset includes eleven distinct named entity types: *(i)* ABSTRACT; *(ii)* ANIMAL; *(iii)* EVENT; *(iv)* OBJECT; *(v)* ORGANIZATION; *(vi)* PERSON; *(vii)* PLACE; *(viii)* PLANT; *(ix)* QUANTITY; *(x)* SUBSTANCE; and *(xi)* TIME.

SKILLSPAN The SKILLSPAN (Zhang et al., 2022a) dataset is composed of a single entity type, SOFTSKILLS, extracted from job descriptions. Unlike traditional entities, soft skills do not follow a fixed syntactic or semantic structure, making them inherently ambiguous.

B Structured Output from LLMs

For a label-sensitive task like NER, getting a structured output from a LLM is a crucial step. In the NER task, as defined in Section 3.1, each token in a sentence is tagged with a corresponding label. Hence, preserving the token-label correspondence in the output is necessary for the LLMs. The most recent LLMs are based on a decoder architecture that, while being suitable for sequence-to-sequence tasks, encounters challenges when tackling the NER task due to the potential misalignment between tokens and labels (Ul Haq et al., 2024). In fact, recent studies on NER (Li et al., 2024; Liu et al., 2024; Wang et al., 2023) have shown that the decoder architecture presents structural inconsistencies in the output. Recently, OpenAI (OpenAI, 2023) released a feature for the latest GPT-4 based models, which guarantees to follow the structured output format[5]. To address the token-label misalignment problem, this study leverages the latest feature of `StructuredOutput`, released by OpenAI. However, it is important to note that despite the inclusion of such features in the latest LLMs, including Qwen (Team, 2024b) and Llama (Touvron et al., 2023) based models, they still exhibit inconsistencies in their output, unlike the `gpt-4o-mini-2024-07-18`.

C Complete Results

This section reports the model's outcomes regarding the experiments performed in this study. Table 5, Table 6, and Table 7 show the performance of models in terms of Precision, Recall, and F_1 score.

[5] https://openai.com/index/introducing-structured-outputs-in-the-api

Table 5: The F_1 score, Precision, and Recall, along with standard deviation, are reported on the test set. The values are averaged over five different random initializations. Bold shows the highest value of F_1 score in each split column-wise. "#Ex." represents the number of context examples used, while "Baseline" refers to the use of LLM with no context examples.

#Ex.	Method	CoNLL2003			WNUT-17			GUM			SKILLSPAN		
		P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$
	Human	$91.09_{\pm 0.49}$	$93.17_{\pm 0.17}$	$92.12_{\pm 0.33}$	$65.21_{\pm 2.32}$	$47.48_{\pm 1.83}$	$54.93_{\pm 1.67}$	$55.07_{\pm 0.31}$	$61.86_{\pm 0.44}$	$58.26_{\pm 0.19}$	$54.30_{\pm 1.60}$	$55.38_{\pm 1.75}$	$54.79_{\pm 0.26}$
	gpt-4o-mini-2024-07-18												
	Baseline	$64.65_{\pm 0.85}$	$80.37_{\pm 0.50}$	$71.66_{\pm 0.41}$	$47.35_{\pm 2.46}$	$55.18_{\pm 2.84}$	$50.88_{\pm 1.14}$	$20.32_{\pm 5.26}$	$13.93_{\pm 2.76}$	$16.42_{\pm 3.34}$	$11.09_{\pm 0.97}$	$17.83_{\pm 2.02}$	$13.59_{\pm 0.52}$
25	ICL	$76.48_{\pm 0.43}$	$82.06_{\pm 0.35}$	$79.17_{\pm 0.25}$	$53.18_{\pm 3.22}$	$52.24_{\pm 2.73}$	$\mathbf{52.58}_{\pm 0.78}$	$44.06_{\pm 0.69}$	$52.04_{\pm 1.57}$	$47.71_{\pm 0.79}$	$21.23_{\pm 1.46}$	$45.26_{\pm 1.72}$	$28.86_{\pm 1.24}$
	RAG w/ST	$84.48_{\pm 1.04}$	$88.99_{\pm 0.65}$	$86.68_{\pm 0.85}$	$51.42_{\pm 2.63}$	$50.98_{\pm 1.52}$	$51.14_{\pm 1.01}$	$46.09_{\pm 0.66}$	$54.38_{\pm 1.07}$	$49.89_{\pm 0.70}$	$20.29_{\pm 0.78}$	$49.47_{\pm 1.80}$	$28.77_{\pm 0.93}$
	RAG w/OpenAI	$87.35_{\pm 0.65}$	$90.71_{\pm 0.34}$	$\mathbf{89.00}_{\pm 0.29}$	$52.26_{\pm 2.24}$	$49.75_{\pm 1.51}$	$50.93_{\pm 0.93}$	$47.04_{\pm 0.23}$	$57.56_{\pm 1.44}$	$\mathbf{51.77}_{\pm 0.66}$	$21.26_{\pm 1.69}$	$56.74_{\pm 1.37}$	$\mathbf{30.91}_{\pm 1.94}$
50	ICL	$79.77_{\pm 0.34}$	$82.64_{\pm 0.49}$	$81.18_{\pm 0.29}$	$55.75_{\pm 2.80}$	$49.53_{\pm 3.07}$	$52.33_{\pm 0.97}$	$45.12_{\pm 0.82}$	$54.35_{\pm 2.02}$	$49.28_{\pm 0.88}$	$20.56_{\pm 0.89}$	$47.42_{\pm 2.01}$	$28.66_{\pm 0.85}$
	RAG w/ST	$86.73_{\pm 1.03}$	$89.29_{\pm 0.84}$	$87.99_{\pm 0.90}$	$53.74_{\pm 3.02}$	$48.74_{\pm 4.44}$	$50.90_{\pm 1.34}$	$46.46_{\pm 1.34}$	$55.46_{\pm 1.21}$	$50.56_{\pm 1.29}$	$22.22_{\pm 1.47}$	$52.60_{\pm 1.41}$	$31.20_{\pm 1.32}$
	RAG w/OpenAI	$87.43_{\pm 0.48}$	$91.39_{\pm 0.16}$	$\mathbf{89.36}_{\pm 0.27}$	$56.53_{\pm 2.35}$	$50.29_{\pm 2.64}$	$\mathbf{53.14}_{\pm 0.75}$	$47.32_{\pm 0.92}$	$58.44_{\pm 1.21}$	$\mathbf{52.28}_{\pm 0.65}$	$23.88_{\pm 1.09}$	$54.28_{\pm 2.26}$	$\mathbf{33.13}_{\pm 0.77}$
75	ICL	$78.74_{\pm 1.02}$	$83.17_{\pm 0.55}$	$80.89_{\pm 0.66}$	$51.90_{\pm 4.29}$	$52.85_{\pm 1.95}$	$52.24_{\pm 1.76}$	$44.40_{\pm 0.63}$	$53.89_{\pm 1.79}$	$48.67_{\pm 0.69}$	$20.84_{\pm 1.59}$	$52.06_{\pm 1.01}$	$29.73_{\pm 1.58}$
	RAG w/ST	$86.91_{\pm 0.31}$	$89.25_{\pm 0.44}$	$88.06_{\pm 0.26}$	$53.80_{\pm 1.75}$	$51.79_{\pm 1.88}$	$52.73_{\pm 0.80}$	$47.22_{\pm 0.98}$	$55.57_{\pm 0.43}$	$51.05_{\pm 0.60}$	$21.39_{\pm 0.87}$	$52.85_{\pm 1.10}$	$30.43_{\pm 0.73}$
	RAG w/OpenAI	$88.07_{\pm 0.35}$	$91.44_{\pm 0.28}$	$\mathbf{89.72}_{\pm 0.25}$	$55.72_{\pm 4.22}$	$51.71_{\pm 3.34}$	$\mathbf{53.43}_{\pm 0.54}$	$47.04_{\pm 1.29}$	$58.19_{\pm 1.18}$	$\mathbf{52.02}_{\pm 1.15}$	$24.66_{\pm 1.34}$	$55.39_{\pm 3.19}$	$\mathbf{34.06}_{\pm 0.88}$
	DeepSeek-R1-Distill-Qwen-32B												
	Baseline	$31.19_{\pm 1.27}$	$63.05_{\pm 0.63}$	$41.72_{\pm 1.21}$	$32.86_{\pm 2.68}$	$39.17_{\pm 3.04}$	$35.70_{\pm 2.47}$	$9.83_{\pm 2.20}$	$5.21_{\pm 2.81}$	$6.35_{\pm 2.40}$	$2.55_{\pm 0.53}$	$4.28_{\pm 1.55}$	$3.18_{\pm 0.85}$
25	ICL	$58.09_{\pm 1.76}$	$80.28_{\pm 1.37}$	$67.39_{\pm 1.37}$	$47.61_{\pm 1.64}$	$58.30_{\pm 2.41}$	$52.37_{\pm 1.12}$	$34.20_{\pm 1.70}$	$44.86_{\pm 1.41}$	$38.78_{\pm 1.09}$	$20.15_{\pm 1.32}$	$43.35_{\pm 0.47}$	$27.49_{\pm 1.25}$
	RAG w/ST	$82.48_{\pm 1.86}$	$88.01_{\pm 1.36}$	$85.15_{\pm 1.27}$	$50.62_{\pm 2.77}$	$55.76_{\pm 1.52}$	$53.03_{\pm 1.79}$	$37.68_{\pm 0.99}$	$47.34_{\pm 1.76}$	$41.94_{\pm 0.90}$	$22.33_{\pm 1.87}$	$47.93_{\pm 1.53}$	$30.42_{\pm 1.75}$
	RAG w/OpenAI	$82.92_{\pm 1.35}$	$90.99_{\pm 0.45}$	$\mathbf{86.76}_{\pm 0.69}$	$51.24_{\pm 2.61}$	$56.78_{\pm 2.12}$	$\mathbf{53.79}_{\pm 0.63}$	$37.68_{\pm 1.38}$	$51.33_{\pm 0.82}$	$\mathbf{43.45}_{\pm 1.12}$	$21.97_{\pm 1.67}$	$53.01_{\pm 2.37}$	$\mathbf{31.04}_{\pm 1.90}$
50	ICL	$62.46_{\pm 0.87}$	$80.28_{\pm 1.05}$	$70.26_{\pm 0.58}$	$49.64_{\pm 3.03}$	$58.26_{\pm 2.77}$	$53.53_{\pm 1.74}$	$34.20_{\pm 1.04}$	$46.42_{\pm 1.21}$	$39.38_{\pm 0.95}$	$20.99_{\pm 0.71}$	$43.25_{\pm 1.70}$	$28.24_{\pm 0.38}$
	RAG w/ST	$81.46_{\pm 0.69}$	$87.92_{\pm 1.06}$	$84.56_{\pm 0.63}$	$52.65_{\pm 3.07}$	$55.46_{\pm 2.71}$	$53.90_{\pm 0.55}$	$38.57_{\pm 0.64}$	$48.96_{\pm 0.99}$	$43.15_{\pm 0.71}$	$21.04_{\pm 0.59}$	$49.23_{\pm 2.05}$	$29.46_{\pm 0.66}$
	RAG w/OpenAI	$85.50_{\pm 0.97}$	$90.70_{\pm 0.19}$	$\mathbf{88.02}_{\pm 0.47}$	$54.14_{\pm 0.89}$	$56.45_{\pm 1.97}$	$\mathbf{55.24}_{\pm 0.58}$	$40.76_{\pm 0.88}$	$50.49_{\pm 1.82}$	$\mathbf{45.09}_{\pm 0.96}$	$24.42_{\pm 2.12}$	$54.95_{\pm 2.43}$	$\mathbf{33.72}_{\pm 1.63}$
75	ICL	$50.04_{\pm 1.00}$	$80.67_{\pm 1.17}$	$61.77_{\pm 1.07}$	$52.17_{\pm 2.72}$	$52.01_{\pm 2.95}$	$51.98_{\pm 0.85}$	$33.61_{\pm 1.99}$	$44.38_{\pm 1.46}$	$38.21_{\pm 1.25}$	$21.63_{\pm 0.82}$	$47.70_{\pm 1.51}$	$29.75_{\pm 0.85}$
	RAG w/ST	$81.85_{\pm 1.49}$	$87.57_{\pm 0.75}$	$84.61_{\pm 0.97}$	$53.11_{\pm 1.17}$	$54.22_{\pm 2.38}$	$53.62_{\pm 0.92}$	$38.63_{\pm 1.08}$	$49.76_{\pm 1.65}$	$43.48_{\pm 1.01}$	$21.21_{\pm 1.75}$	$51.38_{\pm 2.09}$	$29.96_{\pm 1.34}$
	RAG w/OpenAI	$84.87_{\pm 0.56}$	$90.21_{\pm 0.31}$	$\mathbf{87.46}_{\pm 0.38}$	$55.95_{\pm 3.32}$	$54.70_{\pm 1.01}$	$\mathbf{55.26}_{\pm 1.15}$	$38.16_{\pm 1.69}$	$50.61_{\pm 1.32}$	$\mathbf{43.50}_{\pm 1.47}$	$23.43_{\pm 2.00}$	$53.68_{\pm 1.97}$	$\mathbf{32.55}_{\pm 1.74}$

Table 6: The F_1 score, Precision, and Recall, along with standard deviation, are reported on the test set. The values are averaged over five different random initializations. Bold shows the highest value of F_1 score in each split column-wise. "#Ex." represents the number of context examples used, while "Baseline" refers to the use of LLM with no context examples.

#Ex.	Method	CoNLL2003			WNUT-17			GUM			SKILLSPAN		
		P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$
	Human	$91.09_{\pm 0.49}$	$93.17_{\pm 0.17}$	$92.12_{\pm 0.33}$	$65.21_{\pm 2.32}$	$47.48_{\pm 1.83}$	$54.93_{\pm 1.67}$	$55.07_{\pm 0.31}$	$61.86_{\pm 0.44}$	$58.26_{\pm 0.19}$	$54.30_{\pm 1.60}$	$55.38_{\pm 1.75}$	$54.79_{\pm 0.26}$
	Qwen2.5-72B-Instruct												
	Baseline	$26.97_{\pm 0.28}$	$60.80_{\pm 1.25}$	$37.36_{\pm 0.30}$	$16.40_{\pm 1.30}$	$41.19_{\pm 1.23}$	$23.43_{\pm 1.42}$	$6.32_{\pm 0.21}$	$27.46_{\pm 0.97}$	$10.28_{\pm 0.35}$	$4.89_{\pm 0.41}$	$13.79_{\pm 2.15}$	$7.21_{\pm 0.69}$
25	ICL	$74.57_{\pm 0.79}$	$83.57_{\pm 0.82}$	$78.81_{\pm 0.30}$	$45.58_{\pm 2.66}$	$59.47_{\pm 3.09}$	$\mathbf{51.49}_{\pm 0.92}$	$41.69_{\pm 1.10}$	$55.80_{\pm 1.00}$	$47.73_{\pm 1.06}$	$17.06_{\pm 2.18}$	$31.17_{\pm 1.63}$	$22.01_{\pm 2.13}$
	RAG w/ST	$81.87_{\pm 0.72}$	$89.90_{\pm 0.46}$	$85.69_{\pm 0.56}$	$46.55_{\pm 2.70}$	$45.68_{\pm 1.43}$	$46.06_{\pm 1.33}$	$47.79_{\pm 1.05}$	$60.15_{\pm 0.99}$	$\mathbf{53.26}_{\pm 0.89}$	$18.25_{\pm 2.05}$	$47.93_{\pm 2.08}$	$26.40_{\pm 2.37}$
	RAG w/OpenAI	$84.81_{\pm 1.16}$	$91.68_{\pm 0.64}$	$\mathbf{88.11}_{\pm 0.82}$	$48.33_{\pm 2.82}$	$49.88_{\pm 1.89}$	$49.05_{\pm 1.86}$	$47.16_{\pm 0.46}$	$59.83_{\pm 0.64}$	$52.74_{\pm 0.16}$	$18.23_{\pm 1.90}$	$50.07_{\pm 4.49}$	$\mathbf{26.63}_{\pm 1.90}$
50	ICL	$77.48_{\pm 0.51}$	$83.34_{\pm 0.53}$	$80.30_{\pm 0.43}$	$45.04_{\pm 1.78}$	$59.31_{\pm 1.71}$	$51.17_{\pm 1.16}$	$44.30_{\pm 1.09}$	$57.69_{\pm 1.35}$	$50.12_{\pm 1.14}$	$17.51_{\pm 0.85}$	$33.82_{\pm 1.01}$	$23.06_{\pm 0.86}$
	RAG w/ST	$84.30_{\pm 0.98}$	$91.49_{\pm 0.85}$	$87.74_{\pm 0.77}$	$45.60_{\pm 2.82}$	$56.63_{\pm 1.52}$	$50.45_{\pm 1.14}$	$48.83_{\pm 1.45}$	$60.55_{\pm 1.05}$	$\mathbf{54.06}_{\pm 1.25}$	$21.32_{\pm 1.82}$	$55.79_{\pm 4.56}$	$30.84_{\pm 2.52}$
	RAG w/OpenAI	$85.96_{\pm 1.44}$	$92.32_{\pm 0.30}$	$\mathbf{89.02}_{\pm 0.66}$	$48.66_{\pm 2.91}$	$57.09_{\pm 2.42}$	$\mathbf{52.46}_{\pm 1.23}$	$47.33_{\pm 0.81}$	$61.23_{\pm 0.44}$	$53.38_{\pm 0.63}$	$23.44_{\pm 2.19}$	$52.32_{\pm 2.82}$	$\mathbf{32.35}_{\pm 2.54}$
75	ICL	$77.50_{\pm 0.68}$	$83.60_{\pm 0.74}$	$80.43_{\pm 0.67}$	$52.14_{\pm 2.27}$	$55.62_{\pm 3.11}$	$53.72_{\pm 0.80}$	$47.60_{\pm 0.77}$	$57.08_{\pm 1.22}$	$51.91_{\pm 0.80}$	$20.81_{\pm 1.15}$	$48.26_{\pm 2.80}$	$29.05_{\pm 1.26}$
	RAG w/ST	$87.46_{\pm 0.39}$	$91.95_{\pm 0.29}$	$89.65_{\pm 0.31}$	$48.36_{\pm 3.25}$	$55.51_{\pm 1.88}$	$51.58_{\pm 1.04}$	$50.29_{\pm 0.27}$	$60.97_{\pm 0.51}$	$\mathbf{55.11}_{\pm 0.17}$	$20.99_{\pm 2.12}$	$49.99_{\pm 1.23}$	$\mathbf{29.52}_{\pm 2.10}$
	RAG w/OpenAI	$86.77_{\pm 0.54}$	$92.05_{\pm 0.72}$	$\mathbf{89.34}_{\pm 0.61}$	$48.56_{\pm 2.08}$	$60.22_{\pm 1.52}$	$\mathbf{53.72}_{\pm 0.71}$	$47.24_{\pm 1.27}$	$60.34_{\pm 0.57}$	$52.98_{\pm 0.76}$	$19.95_{\pm 0.74}$	$50.74_{\pm 1.47}$	$28.62_{\pm 0.77}$
	Llama3.1-70B-Instruct												
	Baseline	$23.56_{\pm 0.10}$	$63.25_{\pm 0.17}$	$34.33_{\pm 0.15}$	$16.35_{\pm 0.74}$	$54.65_{\pm 0.42}$	$25.16_{\pm 0.84}$	$6.44_{\pm 0.08}$	$27.79_{\pm 0.35}$	$10.46_{\pm 0.13}$	$3.51_{\pm 0.08}$	$24.30_{\pm 0.64}$	$6.14_{\pm 0.12}$
25	ICL	$73.59_{\pm 0.78}$	$78.73_{\pm 1.03}$	$76.06_{\pm 0.41}$	$48.77_{\pm 2.20}$	$47.66_{\pm 5.18}$	$\mathbf{48.00}_{\pm 2.14}$	$18.26_{\pm 2.80}$	$41.83_{\pm 0.98}$	$25.34_{\pm 2.68}$	$17.04_{\pm 0.52}$	$45.86_{\pm 2.86}$	$24.84_{\pm 0.95}$
	RAG w/ST	$83.15_{\pm 1.42}$	$86.37_{\pm 0.90}$	$\mathbf{84.72}_{\pm 0.54}$	$36.68_{\pm 1.32}$	$49.10_{\pm 3.77}$	$41.89_{\pm 0.99}$	$43.09_{\pm 1.10}$	$50.88_{\pm 2.31}$	$\mathbf{46.63}_{\pm 0.89}$	$19.62_{\pm 1.44}$	$46.47_{\pm 1.76}$	$\mathbf{27.55}_{\pm 1.21}$
	RAG w/OpenAI	$68.32_{\pm 3.99}$	$87.50_{\pm 1.82}$	$76.65_{\pm 2.19}$	$43.52_{\pm 4.33}$	$44.71_{\pm 3.86}$	$43.82_{\pm 1.29}$	$42.46_{\pm 1.75}$	$48.87_{\pm 4.60}$	$45.29_{\pm 1.50}$	$19.59_{\pm 1.52}$	$42.16_{\pm 1.49}$	$26.73_{\pm 1.65}$
50	ICL	$76.13_{\pm 1.12}$	$76.79_{\pm 1.24}$	$76.44_{\pm 0.30}$	$50.24_{\pm 2.81}$	$48.90_{\pm 2.24}$	$\mathbf{49.48}_{\pm 1.08}$	$35.67_{\pm 1.83}$	$48.79_{\pm 3.19}$	$41.12_{\pm 1.07}$	$16.09_{\pm 0.97}$	$44.15_{\pm 4.16}$	$23.51_{\pm 0.71}$
	RAG w/ST	$83.87_{\pm 0.69}$	$88.57_{\pm 0.88}$	$\mathbf{86.15}_{\pm 0.28}$	$42.92_{\pm 2.03}$	$48.79_{\pm 2.99}$	$45.57_{\pm 0.76}$	$43.76_{\pm 1.50}$	$50.24_{\pm 2.00}$	$\mathbf{46.73}_{\pm 0.49}$	$17.69_{\pm 0.66}$	$46.11_{\pm 4.63}$	$25.50_{\pm 0.54}$
	RAG w/OpenAI	$68.36_{\pm 1.53}$	$89.08_{\pm 0.75}$	$77.35_{\pm 0.97}$	$44.14_{\pm 1.97}$	$51.28_{\pm 2.94}$	$47.36_{\pm 0.64}$	$43.70_{\pm 2.43}$	$49.70_{\pm 1.83}$	$46.45_{\pm 1.40}$	$18.37_{\pm 2.42}$	$44.41_{\pm 4.44}$	$\mathbf{25.77}_{\pm 1.75}$
75	ICL	$74.94_{\pm 1.03}$	$75.15_{\pm 1.03}$	$75.04_{\pm 0.70}$	$50.78_{\pm 1.74}$	$51.69_{\pm 2.43}$	$\mathbf{51.18}_{\pm 1.05}$	$39.62_{\pm 1.64}$	$47.88_{\pm 3.05}$	$43.30_{\pm 1.39}$	$17.55_{\pm 1.05}$	$51.80_{\pm 1.68}$	$26.19_{\pm 1.14}$
	RAG w/ST	$85.70_{\pm 0.60}$	$89.03_{\pm 0.55}$	$\mathbf{87.33}_{\pm 0.23}$	$47.41_{\pm 3.89}$	$51.36_{\pm 1.97}$	$49.18_{\pm 1.61}$	$45.84_{\pm 1.19}$	$50.36_{\pm 1.12}$	$47.98_{\pm 0.68}$	$18.87_{\pm 1.35}$	$51.17_{\pm 2.06}$	$\mathbf{27.52}_{\pm 1.18}$
	RAG w/OpenAI	$76.99_{\pm 1.57}$	$87.46_{\pm 1.39}$	$81.87_{\pm 0.67}$	$49.43_{\pm 4.27}$	$48.16_{\pm 5.99}$	$48.39_{\pm 2.17}$	$44.46_{\pm 0.61}$	$52.96_{\pm 1.65}$	$\mathbf{48.33}_{\pm 0.64}$	$9.51_{\pm 1.65}$	$47.74_{\pm 3.51}$	$15.83_{\pm 2.47}$

Table 7: The F_1 score, Precision, and Recall, along with standard deviation, are reported on the test set. The values are averaged over five different random initializations. Bold shows the highest value of F_1 score in each split column-wise. "#Ex." represents the number of context examples used, while "Baseline" refers to the use of LLM with no context examples.

#Ex.	Method	CoNLL2003			WNUT-17			GUM			SKILLSPAN		
		P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$	P	R	$\mathbf{F_1}$
	Human	$91.09_{\pm 0.49}$	$93.17_{\pm 0.17}$	$92.12_{\pm 0.33}$	$65.21_{\pm 2.32}$	$47.48_{\pm 1.83}$	$54.93_{\pm 1.67}$	$55.07_{\pm 0.31}$	$61.86_{\pm 0.44}$	$58.26_{\pm 0.19}$	$54.30_{\pm 1.60}$	$55.38_{\pm 1.75}$	$54.79_{\pm 0.26}$
	Qwen2.5-7B-Instruct												
	Baseline	$21.79_{\pm 1.28}$	$62.11_{\pm 0.44}$	$32.24_{\pm 1.44}$	$20.95_{\pm 2.83}$	$44.08_{\pm 3.68}$	$28.36_{\pm 3.17}$	$3.27_{\pm 0.22}$	$14.10_{\pm 1.03}$	$5.31_{\pm 0.37}$	$5.41_{\pm 1.05}$	$35.29_{\pm 2.73}$	$9.35_{\pm 1.58}$
25	ICL	$70.22_{\pm 1.45}$	$75.96_{\pm 1.49}$	$72.95_{\pm 0.30}$	$47.79_{\pm 3.40}$	$47.02_{\pm 2.78}$	$\mathbf{47.29}_{\pm 1.63}$	$28.31_{\pm 1.01}$	$44.01_{\pm 0.97}$	$34.43_{\pm 0.57}$	$14.12_{\pm 0.88}$	$54.89_{\pm 1.48}$	$22.44_{\pm 1.08}$
	RAG w/ST	$83.81_{\pm 0.67}$	$89.68_{\pm 0.57}$	$86.64_{\pm 0.45}$	$37.82_{\pm 3.45}$	$49.68_{\pm 3.12}$	$42.81_{\pm 2.14}$	$35.90_{\pm 1.86}$	$50.09_{\pm 2.04}$	$\mathbf{41.80}_{\pm 1.65}$	$15.99_{\pm 0.38}$	$55.06_{\pm 0.93}$	$24.77_{\pm 0.42}$
	RAG w/OpenAI	$84.05_{\pm 1.15}$	$90.85_{\pm 0.31}$	$\mathbf{87.32}_{\pm 0.65}$	$50.22_{\pm 3.43}$	$41.75_{\pm 4.88}$	$45.30_{\pm 1.73}$	$34.63_{\pm 1.09}$	$49.97_{\pm 1.00}$	$40.89_{\pm 0.52}$	$20.45_{\pm 1.35}$	$54.60_{\pm 4.83}$	$\mathbf{29.67}_{\pm 1.29}$
50	ICL	$72.55_{\pm 1.01}$	$78.54_{\pm 0.32}$	$75.42_{\pm 0.57}$	$47.95_{\pm 3.18}$	$49.36_{\pm 3.94}$	$\mathbf{48.54}_{\pm 2.51}$	$33.51_{\pm 0.76}$	$43.59_{\pm 1.18}$	$37.88_{\pm 0.56}$	$15.13_{\pm 0.90}$	$52.64_{\pm 2.98}$	$23.47_{\pm 1.01}$
	RAG w/ST	$85.78_{\pm 0.69}$	$90.21_{\pm 0.38}$	$\mathbf{87.94}_{\pm 0.43}$	$52.14_{\pm 4.79}$	$44.00_{\pm 3.95}$	$47.41_{\pm 0.49}$	$39.38_{\pm 1.31}$	$49.85_{\pm 1.69}$	$\mathbf{43.97}_{\pm 0.44}$	$17.36_{\pm 1.08}$	$51.06_{\pm 2.65}$	$25.87_{\pm 1.02}$
	RAG w/OpenAI	$80.90_{\pm 1.79}$	$91.55_{\pm 0.36}$	$85.89_{\pm 1.13}$	$41.97_{\pm 2.87}$	$48.62_{\pm 5.64}$	$44.75_{\pm 0.98}$	$34.63_{\pm 1.32}$	$50.61_{\pm 1.67}$	$41.11_{\pm 1.40}$	$18.12_{\pm 0.94}$	$56.68_{\pm 3.93}$	$\mathbf{27.41}_{\pm 0.73}$
75	ICL	$81.36_{\pm 1.19}$	$75.72_{\pm 1.00}$	$78.43_{\pm 0.63}$	$47.90_{\pm 5.76}$	$47.78_{\pm 3.57}$	$47.51_{\pm 1.97}$	$34.23_{\pm 2.14}$	$46.40_{\pm 1.10}$	$39.39_{\pm 1.77}$	$12.98_{\pm 1.17}$	$51.23_{\pm 6.20}$	$20.68_{\pm 1.81}$
	RAG w/ST	$86.54_{\pm 1.93}$	$88.40_{\pm 1.15}$	$\mathbf{87.44}_{\pm 0.87}$	$52.39_{\pm 4.73}$	$47.17_{\pm 1.99}$	$\mathbf{49.48}_{\pm 1.30}$	$40.14_{\pm 1.39}$	$48.15_{\pm 1.09}$	$43.76_{\pm 0.73}$	$18.34_{\pm 0.46}$	$46.32_{\pm 3.08}$	$\mathbf{26.25}_{\pm 0.76}$
	RAG w/OpenAI	$81.67_{\pm 1.51}$	$90.96_{\pm 0.30}$	$86.06_{\pm 0.88}$	$48.73_{\pm 1.31}$	$47.85_{\pm 2.49}$	$48.25_{\pm 1.30}$	$39.56_{\pm 1.25}$	$50.87_{\pm 1.25}$	$\mathbf{44.48}_{\pm 0.55}$	$14.07_{\pm 0.80}$	$61.07_{\pm 0.86}$	$22.86_{\pm 1.06}$
	Llama-3.1-8B-Instruct												
	Baseline	$22.98_{\pm 0.67}$	$74.87_{\pm 0.48}$	$35.17_{\pm 0.83}$	$11.06_{\pm 2.70}$	$36.38_{\pm 10.40}$	$16.88_{\pm 4.16}$	$6.98_{\pm 0.03}$	$28.22_{\pm 0.18}$	$11.19_{\pm 0.05}$	$3.03_{\pm 0.21}$	$20.37_{\pm 5.76}$	$5.22_{\pm 0.32}$
25	ICL	$63.86_{\pm 0.95}$	$75.71_{\pm 1.61}$	$69.26_{\pm 0.69}$	$35.94_{\pm 3.54}$	$51.58_{\pm 2.70}$	$\mathbf{42.23}_{\pm 2.38}$	$33.95_{\pm 1.97}$	$41.74_{\pm 3.02}$	$37.39_{\pm 1.85}$	$12.40_{\pm 0.85}$	$33.63_{\pm 6.65}$	$17.93_{\pm 0.66}$
	RAG w/ST	$78.44_{\pm 1.18}$	$86.16_{\pm 0.88}$	$\mathbf{82.11}_{\pm 0.86}$	$36.82_{\pm 4.64}$	$43.86_{\pm 7.22}$	$39.38_{\pm 2.26}$	$39.94_{\pm 2.23}$	$46.48_{\pm 0.96}$	$42.92_{\pm 1.20}$	$14.95_{\pm 1.88}$	$42.25_{\pm 6.45}$	$\mathbf{21.87}_{\pm 1.51}$
	RAG w/OpenAI	$69.03_{\pm 1.02}$	$86.41_{\pm 2.27}$	$76.73_{\pm 1.02}$	$32.83_{\pm 3.20}$	$48.82_{\pm 7.41}$	$38.89_{\pm 2.78}$	$40.77_{\pm 1.82}$	$49.07_{\pm 2.80}$	$\mathbf{44.45}_{\pm 0.54}$	$12.16_{\pm 0.97}$	$41.55_{\pm 3.20}$	$18.79_{\pm 1.31}$
50	ICL	$67.78_{\pm 1.48}$	$76.79_{\pm 0.69}$	$72.01_{\pm 1.13}$	$40.49_{\pm 1.76}$	$48.82_{\pm 2.88}$	$\mathbf{44.20}_{\pm 1.03}$	$36.43_{\pm 1.51}$	$42.53_{\pm 2.12}$	$39.22_{\pm 1.32}$	$12.94_{\pm 1.13}$	$35.45_{\pm 2.12}$	$18.90_{\pm 1.01}$
	RAG w/ST	$79.29_{\pm 3.86}$	$86.85_{\pm 2.00}$	$\mathbf{82.82}_{\pm 1.41}$	$40.04_{\pm 4.26}$	$48.60_{\pm 4.26}$	$43.59_{\pm 1.04}$	$39.73_{\pm 1.81}$	$46.54_{\pm 2.12}$	$42.81_{\pm 0.99}$	$15.13_{\pm 0.96}$	$47.13_{\pm 1.48}$	$\mathbf{22.88}_{\pm 0.99}$
	RAG w/OpenAI	$69.98_{\pm 1.50}$	$87.05_{\pm 2.15}$	$77.56_{\pm 0.74}$	$39.75_{\pm 1.82}$	$49.53_{\pm 2.63}$	$44.03_{\pm 0.76}$	$40.89_{\pm 1.62}$	$46.96_{\pm 2.27}$	$\mathbf{43.66}_{\pm 0.67}$	$12.09_{\pm 1.18}$	$42.63_{\pm 3.13}$	$18.76_{\pm 1.19}$
75	ICL	$71.44_{\pm 2.02}$	$77.86_{\pm 2.41}$	$74.47_{\pm 1.00}$	$39.13_{\pm 1.16}$	$48.70_{\pm 2.37}$	$43.35_{\pm 0.84}$	$34.33_{\pm 1.27}$	$41.30_{\pm 2.37}$	$37.43_{\pm 0.53}$	$13.37_{\pm 1.12}$	$37.83_{\pm 4.68}$	$19.67_{\pm 1.20}$
	RAG w/ST	$82.36_{\pm 2.15}$	$87.69_{\pm 1.70}$	$\mathbf{84.91}_{\pm 0.88}$	$39.92_{\pm 2.09}$	$50.34_{\pm 3.12}$	$\mathbf{44.42}_{\pm 0.59}$	$41.14_{\pm 1.47}$	$43.71_{\pm 3.36}$	$42.30_{\pm 1.57}$	$12.77_{\pm 0.10}$	$45.34_{\pm 0.80}$	$\mathbf{19.92}_{\pm 0.08}$
	RAG w/OpenAI	$74.58_{\pm 2.51}$	$85.21_{\pm 0.99}$	$79.51_{\pm 1.02}$	$41.85_{\pm 2.52}$	$47.17_{\pm 6.92}$	$43.96_{\pm 2.54}$	$41.99_{\pm 1.01}$	$46.13_{\pm 2.68}$	$\mathbf{43.91}_{\pm 0.93}$	$10.42_{\pm 0.96}$	$49.98_{\pm 3.64}$	$17.22_{\pm 1.29}$

D Qualitative Analysis

This study broadly explores the efficacy of LLMs for data annotation tasks, considering different datasets of varying complexity. From the heatmaps visible in Figure 1 of the main article, it is visible that the performance of LLMs decreases as dataset complexity increases. The performance of LLMs on the SKILLSPAN dataset is significantly lower than human annotation, suggesting that even the latest available LLMs struggle to annotate data when the task is complex. For instance, soft skills lack clear or distinct definitions, making the task more challenging. Similarly, the GUM dataset also poses challenges for LLMs due to its entity diversity. On the other hand, in the case of the WNUT-17 and CoNLL-2003 datasets, which consist of simpler entities, annotations are easier to extract for an LLM given its prior knowledge. Furthermore, the quality of context in LLMs plays a major role, particularly in data annotation tasks, where the RAG-based approach significantly outperforms its counterpart. Moreover, for simpler datasets, the RAG-based approach achieves performance comparable to human annotation.

To gain better insights into the performance of the proposed RAG-based approach, Table 8 presents the qualitative results for the SKILLSPAN dataset annotated by `gpt-4o-mini`. In this dataset, data annotation performance remains far below human-level, suggesting that the LLM struggles to extract sufficient information from the context examples when the task is difficult. From Tables 5, 6, and 7, it is observed that LLM-generated annotations improve Recall, whereas Precision is compromised. Table 8 shows that in examples 1 and 4, the LLM incorrectly annotates soft skills that are not identified by human annotators, whereas in examples 2 and 3, the annotations are nearly identical to human annotations. In Example 5, the RAG-based approach performs comparably to human annotation, while both the baseline and ICL fail to do so.

Table 8: Qualitative analysis of soft skills annotations on dataset samples using `gpt-4o-mini-2024-07-18`. The output of the best-performing model is reported. The highlighted texts in the first column are gold labels, while those in the other columns are the corresponding LLM-generated annotations.

№	Human	Baseline	ICL	RAG
1.	Very good understanding of test automation frameworks.	Very good understanding of test automation frameworks.	Very good understanding of test automation frameworks.	Very good understanding of test automation frameworks.
2.	Must have excellent verbal and written skills being able to communicate effectively on both a technical and business level Ability to work under pressure to resolve issues affecting the production services.	Must have excellent verbal and written skills being able to communicate effectively on both a technical and business level Ability to work under pressure to resolve issues affecting the production services.	Must have excellent verbal and written skills being able to communicate effectively on both a technical and business level Ability to work under pressure to resolve issues affecting the production services.	Must have excellent verbal and written skills being able to communicate effectively on both a technical and business level Ability to work under pressure to resolve issues affecting the production services.
3.	Must have excellent work ethic and be detail oriented and be able to work independently.	Must have excellent work ethic and be detail oriented and be able to work independently.	Must have excellent work ethic and be detail oriented and be able to work independently.	Must have excellent work ethic and be detail oriented and be able to work independently.
4.	Technical Skills Core Java.	Technical Skills Core Java.	Technical Skills Core Java.	Technical Skills Core Java.
5.	You will work with the business to define requirements and have excellent communication skills to interpret these into consolidated development scopes.	You will work with the business to define requirements and have excellent communication skills to interpret these into consolidated development scopes.	You will work with the business to define requirements and have excellent communication skills to interpret these into consolidated development scopes.	You will work with the business to define requirements and have excellent communication skills to interpret these into consolidated development scopes.

E Token-level Assessment

This section presents example cases illustrating both agreement and disagreement between LLM-generated annotations and human annotations. The examples are drawn from datasets annotated using `gpt-4o-mini`, RAG, and OpenAI embeddings. For each dataset, four randomly selected examples are visualized: two where the LLM and human annotations align, and two where they differ. Distinct color schemes are used to represent different classes within the same dataset, allowing for clear visual comparison.

CONLL2003 Agreement

Example 1
5. Armstrong 81
LLM Annot.: O, B-PER, O
Human Annot.: O, B-PER, O

Example 2
The man was due to catch a flight to Nuku'alofa in Tonga later on Tuesday .
LLM Annot.: O, O, O, O, O, O, O, O, O, B-LOC, O, B-LOC, O, O, O, O
Human Annot.: O, O, O, O, O, O, O, O, O, B-LOC, O, B-LOC, O, O, O, O

CONLL2003 Disagreement

Example 1
The poll of registered voters showed a shift in favor of the Democrats in the elections for House of Representatives , with 51 percent saying that if the vote were today they would go for a Democrat and 41 percent opting for a Republican .
LLM Annot.: O, O, O, O, O, O, O, O, O, O, O, O, B-ORG, O, O, O, O, B-ORG, I-ORG, I-ORG, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-ORG, O, O, O, O, O, O, B-ORG, O
Human Annot.: O, O, O, O, O, O, O, O, O, O, O, O, B-MISC, O, O, O, O, B-ORG, I-ORG, I-ORG, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-MISC, O, O, O, O, O, O, B-MISC, O

Example 2
Rifkind said it was for the home secretary (interior minister) to act by denying visas to participants if he felt there was reason to believe that they might break the law .
LLM Annot.: B-PER, O, O, O, O, O, B-MISC, I-MISC, O, B-MISC, I-MISC, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O
Human Annot.: B-PER, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

WNUT-17 Agreement

Example 1
Bouncing my ass like thrs no tomorrow
LLM Annot.: O, O, O, O, O, O, O
Human Annot.: O, O, O, O, O, O, O

Example 2
Made it back home to GA . It sucks not to be at Disney world , but its good to be home . Time to start planning the next Disney World trip .
LLM Annot.: O, O, O, O, O, B-location, O, O, O, O, O, O, O, B-location, I-location, O, O, O, O, O, O, O, O, O, O, O, O, O, B-location, I-location, O, O
Human Annot.: O, O, O, O, O, B-location, O, O, O, O, O, O, O, B-location, I-location, O, O, O, O, O, O, O, O, O, O, O, O, O, B-location, I-location, O, O

WNUT-17 Disagreement

Example 1

Via @travelmail Sweden 's ice hotel is celebrating its 25th season and is open for business http://t.co/iE1QN8qEte #travel #tban

LLM Annot.: O, O, B-location, O, B-creative-work, I-creative-work, O, O, O, O, O, O, O, O, O, O, O, O, O

Human Annot.: O, O, B-location, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

Example 2

RT @Sexstrology : How many Sagittarians does it take to change a lightbulb ? The sun is shining , the day is young and youre worrying about ...

LLM Annot.: O, O, O, O, O, B-group, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

Human Annot.: O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

GUM Agreement

Example 1

That is a Category 3 storm

LLM Annot.: B-event, O, B-event, I-event, I-event, I-event

Human Annot.: B-event, O, B-event, I-event, I-event, I-event

Example 2

The 20th century however , marked the rapid development of Athens

LLM Annot.: B-time, I-time, I-time, O, O, O, B-abstract, I-abstract, I-abstract, I-abstract, I-abstract

Human Annot.: B-time, I-time, I-time, O, O, O, B-abstract, I-abstract, I-abstract, I-abstract, I-abstract

GUM Disagreement

Example 1

Also on Main St. is Metropolitan Plants , a haven for flower lovers , with a very large selection of products for gardeners

LLM Annot.: O, O, B-place, I-place, O, B-organization, I-organization, O, B-place, I-place, I-place, I-place, I-place, O, O, B-object, I-object, I-object, I-object, I-object, I-object, O, B-person

Human Annot.: O, O, B-place, I-place, O, B-place, I-place, O, B-place, I-place, I-place, I-place, I-place, O, O, B-object, I-object, I-object, I-object, I-object, I-object, I-object, I-object

Example 2

For example , husbands are likely to laugh at jokes about wives and vice versa ; Purple basil is often grown for decoration , as well as its scent and flowers

LLM Annot.: O, O, O, B-person, O, O, O, O, O, B-event, I-event, B-person, O, O, O, O, B-plant, I-plant, O, O, O, O, B-abstract, O, O, O, O, B-abstract, I-abstract, I-abstract, I-abstract

Human Annot.: O, O, O, B-person, O, O, O, O, O, B-event, I-event, I-event, O, O, O, O, B-plant, I-plant, O, O, O, O, O, O, O, O, O, B-abstract, I-abstract, O, B-plant

SKILLSPAN Agreement

Example 1
Engineering degree (Process Chemical or Mechanical) .
LLM Annot.: O, O, O, O, O, O, O, O, O
Human Annot.: O, O, O, O, O, O, O, O, O

Example 2
Applications submitted through channels other than the online application form below or applications which are not received by the application deadline will not be considered .
LLM Annot.: O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O
Human Annot.: O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

SKILLSPAN Disagreement

Example 1
By `introducing the latest technology` we start `building extremely efficient` even on high load the system of specialised engines .
LLM Annot.: O, `B-Skill`, `I-Skill`, `I-Skill`, `I-Skill`, O, O, `B-Skill`, `I-Skill`, `I-Skill`, O, O, O, O, O, O, O, O, O, O
Human Annot.: O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

Example 2
The company appreciate their employees `enthusiasm` and `work effort` which also means that your `will to work` is highly appreciated .
LLM Annot.: O, O, O, O, O, `B-Skill`, O, `B-Skill`, `I-Skill`, O, O, O, O, O, `B-Skill`, `I-Skill`, `I-Skill`, O, O, O, O
Human Annot.: O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O

F Statistical Significance Test

This study evaluated various large language models across multiple datasets, considering different embeddings and examples as context. While some models clearly outperformed others in the results, the differences in predictions might not be statistically significant for certain models. Therefore, to determine the statistical significance of our findings, a non-parametric test is conducted. This test helps to assess whether there are significant differences among the models and, if so, identify which models differ statistically from each other.

The Friedman test (Pereira et al., 2015) is a non-parametric statistical test used to detect differences in performance across multiple related samples — in this case, different models evaluated over multiple datasets. It ranks the performance scores among datasets and assesses whether the rank distributions differ significantly among models. Let N be the number of datasets, K the number of models, and R_j be the sum of ranks for each model j. The Friedman test statistic chi_F^2, which follows a chi-square distribution, is calculated as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)} \sum_{j=1}^{k} R_j^2 - 3N(k+1). \quad (1)$$

If the test statistic exceeds the critical value for a significance level $\alpha = 0.01$, then the null hypothesis should be rejected, indicating that there are significant differences in performance among the models. If significant differences are found, the post-hoc Conover (Conover, 1999) test is performed to discover pair-wise statistical differences among models while adjusting for multiple comparisons. This test evaluates whether specific models differ significantly in performance.

Given that the Friedman test produces a test statistic of 119.69 with a p-value of 1.76^{-17}, the null hypothesis is rejected, suggesting that at least one model shows a statistically significant difference in ranking. Consequently, the post-hoc Conover test is performed. Figure 2 presents the statistical significance of the model rankings, with significant pairwise differences highlighted accordingly. The x-axis indicates the average rank of each model, where lower ranks closer to the left signify better performance. Each

colored node corresponds to a particular model, labeled with its respective rank, while the black horizontal bars connecting multiple nodes highlight groups of models that do not show statistically significant differences at the specified confidence level. The top-performing combination is `gpt4omini-OpenAI`, with an average rank of 2.8, indicating it consistently outperformed other approaches. Other strong performers include `Qwen2.5-72B-OpenAI` (4.2), `DeepSeek-R1-Distill-Qwen-32B-OpenAI` (4.3), `gpt-4o-mini-ST` (4.9), and `Qwen2.5-72B-ST` (5.8). The models having lower ranks are clustered towards the left. In contrast, `Llama3.1-8B-ICL` (17), `Llama3.1-8B-OpenAI` (15), and `Qwen2.5-7B-ICL` (14) have the highest ranks, suggesting they performed the worst in comparison. These models do not overlap with the higher-ranked ones, highlighting their statistically inferior performance. Interestingly, `Llama3.1-8B-ST` shows no statistical differences when compared to `llama3.1-70B`, whether using ICL or RAG with OpenAI embedding. Similarly, `Qwen2.5-7B`, when utilizing RAG with either OpenAI or ST embeddings, exhibits no statistical differences compared to `llama3.1-70B` using ST embeddings and `Qwen2.5-72B` using ICL. Likewise, `DeepSeek-R1-Distill-Qwen-32B-OpenAI` demonstrates comparable performance against larger variants such as `gpt4omini-OpenAI` and `Qwen2.5-72B`. These tests highlight a crucial aspect: a trade-off when addressing the NER task. Indeed, larger models, such as those with 70B parameters, may not necessarily offer better performance than smaller models like Llama3.1-8B-ST or `Qwen2.5-7B`. This suggests that the additional computational resources required for bigger models might not always justify their use, especially if smaller models can achieve statistically similar results.

G Prompt

This section presents the prompts used to generate the responses of LLMs. These prompts are carefully synthesized to encompass all the components required to get structured output for both: *(i)* baseline, and *(ii)* in-context learning models.

It is important to note that the LLMs adopted in this work support large context windows (> 128K tokens), which enables the inclusion of the full prompt text, including all in-context examples, without exceeding the maximum token budget, i.e., without triggering input truncation or omitting parts of the prompt. In the experiments, the maximum number of in-context examples is 75, corresponding to approximately 500 tokens, which is therefore well within the available context window and ensures that the complete input is always provided to the model.

Baseline Prompt Structure

Task Description
You are an advanced Named-Entity Recognition (NER) system.
Your task is to analyze the given sentence or passage, identify, extract, and classify specific named entities according to the following predefined entity types:

- {labels}

For each sentence:

- **Label** each word in the text with the appropriate entity type if it matches the specified categories.
- Extract **multiple entities** of the same class if they exist.

The output should be in **valid JSON format**, with each word and its corresponding label as shown below.
Follow the structure strictly and do not add any other explanation.
In entities, label the word exactly as in the text. All the text is case-sensitive.

Input

{input_text}

Context Prompt Structure

Task Description
You are an advanced Named-Entity Recognition (NER) system.
Your task is to analyze the given sentence or passage, identify, extract, and classify specific named entities according to the following predefined entity types:

- {labels}

For each sentence:

- **Label** each word in the text with the appropriate entity type if it matches the specified categories.
- Extract **multiple entities** of the same class if they exist.

The output should be in **valid JSON format**, with each word and its corresponding label as shown below.
Follow the structure strictly and do not add any other explanation.
In entities, label the word exactly as in the text. All the text is case-sensitive.

Examples

{context_examples}

Input

{input_text}