LLMs as Data Annotators: How Close Are We to Human Performance

Anonymous ACL submission

Abstract

In NLP, fine-tuning LLMs is effective for various applications but requires high-quality annotated data. However, manual annotation of data is labor-intensive, time-consuming, and costly. Therefore, LLMs are increasingly used to automate the process, often employing in-context learning (ICL) in which some examples related to the task are given in the prompt for better performance. However, manually selecting context examples can lead to inefficiencies and suboptimal model performance. This paper presents comprehensive experiments comparing several LLMs, considering different embedding models, across various datasets for the Named Entity Recognition (NER) task. The evaluation encompasses models with approximately 7B and 70B parameters, including both proprietary and non-proprietary models. Furthermore, leveraging the success of Retrieval-Augmented Generation (RAG), it also considers a method that addresses the limitations of ICL by automatically retrieving contextual examples, thereby enhancing performance. The results highlight the importance of selecting the appropriate LLM and embedding model, understanding the trade-offs between LLM sizes and desired performance, and the necessity to direct research efforts towards more challenging datasets. The code, submitted as Supplementary Material, will be made publicly available after acceptance.

1 Introduction

Data annotation plays a crucial role in training machine learning (ML) models, especially in the era of Natural Language Processing (NLP). 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 data annotation carries even more significance for fine-grained NLP tasks like token classification, where each token of a sentence has to be tagged with a gold label. In specialized domains such as Human Resource Management (HRM) or medical, organizations often possess large datasets that can be leveraged to enhance decision-making and operational efficiency through the use of LLM-based NLP approaches (Urlana et al., 2024). However, for these organizations to fully harness the power of LLMs through fine-tuning, they need high-quality annotated datasets. Traditional data annotation is a labor-intensive and costly process, especially when applied to large corpora. For example, in the case of HRM, annotating a dataset of $10,000$ resumes for an information extraction task can be prohibitively time-consuming and requires significant human effort (Feng et al., 2021).

Nowadays, pre-trained LLMs (Devlin et al., 2019; Liu et al., 2019) can be cost-effectively fine-tuned on downstream tasks. These fine-tuned models are frequently used in scenarios where continuous LLM usage for inference is too expensive, such as when using API provided by propriety services (OpenAI, 2023; Team, 2024a), or when there is the need for tailored models to meet strict performance standards while maintaining the privacy of sensitive information, such as in specialized fields (Strohmeier, 2022; Karabacak and Margetis, 2023). With the advent of advanced LLMs such as GPT-4 (OpenAI, 2023), Qwen (Team, 2024b), and Llama (Touvron et al., 2023), researchers and practitioners are increasingly leveraging these models to enhance the data annotation process (Tan et al., 2024). Pre-trained on massive corpora, LLMs offer unprecedented capabilities for automating and streamlining annotation, improving scalability, and reducing costs (Wang et al., 2021).

Recent studies have demonstrated that LLMs (Wang et al., 2023; Naraki et al., 2024) can achieve performance comparable to human level in

data annotation for Named Entity Recognition Task (NER). However, most of these evaluations are conducted on widely used benchmark datasets such as CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and WNUT-17 (Derczynski et al., 2017). For instance, between 2023 and 2025, the CoNLL-2003 dataset has been utilized in 191 studies, while WNUT-17 has been considered in 45. In contrast, more complex datasets like SKILLSPAN (Zhang et al., 2022a) and GUM (Zeldes, 2017) have been used significantly less frequently, in only 9 and 4 studies[1], respectively. The results presented in this paper suggest that to gain a more comprehensive understanding of model performance regarding data annotation via LLMs in NER task, it is crucial to extend evaluations to more challenging datasets, which better reflect the complexities of real-world applications.

From a technical perspective, in the recent literature, prompting (He et al., 2024a) and in-context learning (ICL) (Dong et al., 2024) are common approaches to leverage the LLMs for data annotation (Tan et al., 2024). ICL, which is a technique where some solved examples of the task are given within the prompt for better7 performance, is generally proven to be more effective. However, selecting the right and relevant examples to use as context for LLMs continues to be a challenging task (Zhang et al., 2022b). Manually choosing examples for each query creates labor overhead, and more significantly, the use of incorrect context examples may lead LLMs to produce hallucinations (Yao et al., 2024) or inaccurate outputs.

To address the above mentioned challenges, this paper presents the following contributions:

1. **Comprehensive Evaluation of LLMs and Embeddings.** It provides a comprehensive assessment of LLMs for data annotation in NER tasks, examining two distinct embedding models, as well as different techniques such as ICL and RAG, while utilizing datasets of varying complexity. It compares five models including proprietary models, such as `gpt-4o-mini`, and open-source alternatives with approximately 7B and 70B parameters scale.

2. **Trade-off Between LLM Sizes and Performances.** The trade-off between LLM sizes and performance is demonstrated, which is further verified by the statistical tests. In fact, with the appropriate LLM and embedding models, there are no statistically significant differences in results between certain 7B and 70B models.

3. **A RAG-Based Annotation Approach.** To improve annotation quality and address the limitations of manual context selection in ICL, this paper considers a RAG based approach (Lewis et al., 2020). Instead of manually crafting in-context examples, the proposed method retrieves the most relevant samples based on similarity scores, enabling LLMs to generate more accurate annotations.

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. The authors evaluated the quality of data generated by the GPT-3 against the human-labeled data. For each sentence to be annotated by the model, they construct a prompt consisting of several human-labeled examples along with the target sentence. They evaluate the performance in n-shot settings. Also, the authors report the performance of text classification and data generation tasks. Likewise, He et al. (2024a) leveraged the use of GPT-3.5 based models to annotate data. In comparison to the previous approach presented by Wang et al. (2021), the authors introduced the concept of chain-of-thought (CoT) (Wei et al., 2023) reasoning to annotate data. The authors simulate the human reasoning process to induce GPT-3.5 to motivate the annotated examples. They present the task description, specific examples, and the corresponding gold labels to GPT-3.5, and then ask the model to explain whether/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.

To leverage the GPT model for the Named Entity Recognition (NER) task, Wang et al.

[1]The statistics regarding the datasets usage is collected from https://paperswithcode.com/dataset/

(2023) proposed a GPT-NER model. The main contribution introduced by the authors is to transform the NER into a text-generation task. The authors used prompt engineering, where prompts consist of three parts: (*i*) task description; (*ii*) few-shot examples; and (*iii*) input sentence. To choose few-shot context examples, they used two different strategies: (*i*) random retrieval; and (*ii*) k-NN based retrieval from training data.

In this work, the authors propose a retrieval-based approach for selecting context examples. Specifically, for each training instance, the method iterates through all tokens in a sentence to identify the k-nearest neighbor (k-NN) tokens. The top k retrieved tokens are then selected, and their corresponding sentences are used as context. The context examples are retrieved from the entire training dataset. Furthermore, for sentences containing multiple entities, the algorithm runs multiple times to ensure the extraction of all entities within the sentence.

Following the work of Wang et al. (2021) and Wang et al. (2023), Naraki et al. (2024) also proposed a LLMs based annotation for 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 demonstrate a procedure that consists of annotating raw data with an LLM to train a task-specific foundation model for NER. Goel et al. (2023) uses the same concept of data annotation using LLMs, however, they do a case study on a medical domain where they leverage the LLMs for accelerating the annotation process along with human input.

The research discussed above highlights the strong interest in using LLMs for dataset annotation, with most approaches relying on ICL. However, systematic evaluation on complex datasets remains limited, and selecting appropriate context examples for ICL is still a challenge. This study provides a comprehensive evaluation of LLMs for NER data annotation.

3 Methodology

3.1 Problem Definition

Given 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 $\mathcal{D}_{test}$. We divide $\mathcal{D}_{train}$ into two disjoint subsets: $\mathcal{X}$ (we call as sample space), from which we sample context examples, 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 $\mathcal{X}$, we select m examples, 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 $\mathcal{T}$.

The NER task can be defined as the problem of learning an approximation function $\widetilde{f_\theta}$ that closely matches the real function $f : \mathcal{S_V} \times \mathcal{V} \rightarrow \mathcal{C}$, where $\mathcal{S_V}$ represents the set of all the possible sentences composed only by words w in the vocabulary $\mathcal{V}$, and $\mathcal{C}$ represents the set of possible entity categories. The real function f given: (*i*) a sentence $S_i \in \mathcal{S_V}$, and (*ii*) a word $w \in \mathcal{V}$, assigns w to its corresponding category $c \in \mathcal{C}$.

3.2 Data Annotation via LLMs

The methodology adopted in the proposed RAG approach is shown in Figure 1. This section discusses the steps followed in the proposed study. Section 3.2.1 explains the prompt template formation, while Section 3.2.2 presents the baseline approach, followed by ICL method in Section 3.2.3. Section 3.2.4 presents the proposed RAG technique, whereas the importance of structured outputs for NER task is discussed in Section 3.2.5.

3.2.1 Prompt Formation

In NLP, crafting an effective prompt for LLMs is a crucial task, as an ill-formed prompt could lead to poor performance. Different LLMs, whether open-source or proprietary, tend to respond differently to variations in prompt (Errica et al., 2024). This work adopts a similar approach to prompt design presented in (He et al., 2024b; Wang et al., 2023), i.e. structuring our prompts around three key components, also visible in Figure 1: (*i*) *Task Description*. This component clearly defines the task the LLM is expected to perform; (*ii*) *Context*. This component provides task-related examples that help the LLM to better understand the problem, while also clarifying the expected input/output

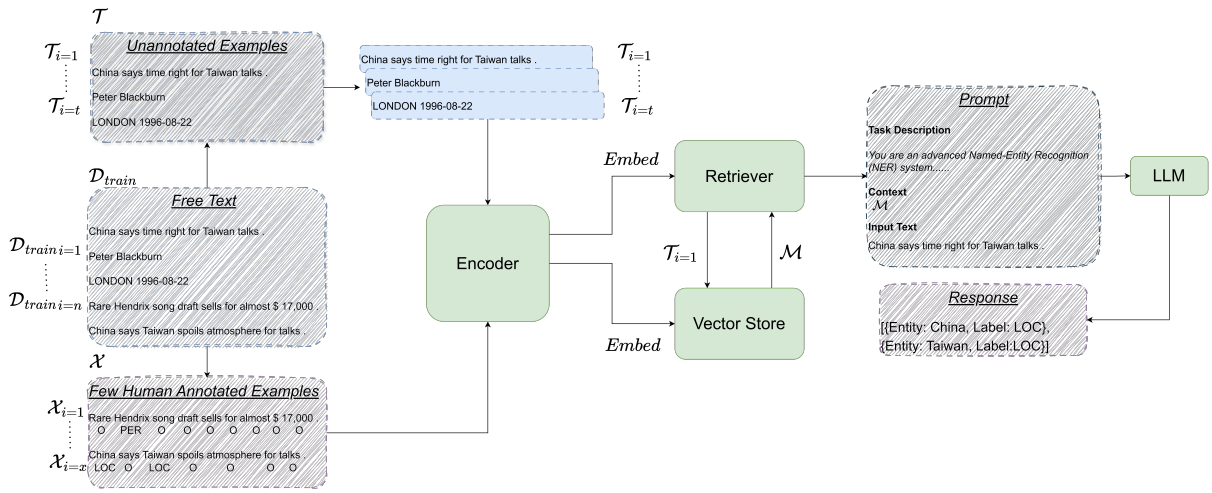

Figure 1: Workflow of the proposed approach. $\mathcal{D}_{train}$ denotes the training data, $\mathcal{X}$ denotes the few human annotated examples, whereas $\mathcal{T}$ denotes the training instances to be annotated by LLM. For each entry $\mathcal{T}_i \in \mathcal{T}$, we extract $\mathcal{M}$ context examples from a vector store using a retriever module. Then, given an input sentence, the final prompt to LLM consists of the task description, the context examples in $\mathcal{M}$, and input sentence.

format; and *(iii) Input*. This final component presents the LLM with the specific examples to be annotated. The prompt structures adopted in the experiments are outlined in Appendix G, while several prompt examples are reported in Appendix H.

3.2.2 Zero-shot Data Annotation

In the zero-shot setting (refers to the baseline), the LLM receives only task descriptions and entity categories from the dataset. The task description explains the task, whereas entity categories provide information about the classes that the LLM has to use for annotation. Providing entity categories in the prompt allows the LLM to produce consistent output annotation as in the training set. For instance, in the CoNLL-2003 dataset, person and organization categories are labelled as PER and ORG respectively. Thus, the prompt to the LLM includes PER and ORG to annotate entities in the person and organization categories, respectively. However, in zero-shot data annotation, the lack of context examples hinders the model's understanding, often leading to suboptimal performance. Nonetheless, this setting allows to evaluate the general knowledge of LLM on a task.

3.2.3 In-Context Learning

In ICL, the prompts given to LLMs are enhanced by including not only a task description and entity categories but also contextual examples. These examples aid the models in better understanding the task at hand. As detailed in Section 3.1, $\mathcal{D}_{train}$ is is split into $\mathcal{X}$ and $\mathcal{T}$. From $\mathcal{X}$, the selection of $\mathcal{M}$ can be approached in two ways: either through manual cherry-picking or by random sampling. However, manually selecting $\mathcal{M}$ can be both time-consuming and subjective, which contradicts the rationale of the proposed study. Therefore, we opt to randomly sample $\mathcal{M}$ from $\mathcal{X}$, although it does not guarantee whether the selected context examples $\mathcal{M}$ are semantically close to the input text $\mathcal{T}_i$, which is a limitation of this approach.

3.2.4 Retrieval-Based Approach

To overcome the limitations of the previously mentioned approaches, this paper introduces a retrieval-based method for automatically selecting relevant context examples. As outlined in Section 3.1, the proposed RAG-based approach first generates embedding representations for all examples in $\mathcal{X}$, which are then stored in a vector database (Douze et al., 2024) for subsequent retrieval, as illustrated in Figure 1. Subsequently, for each sentence $\mathcal{T}_i \in \mathcal{T}$, its embedding representation is generated, and the most similar $\mathcal{M}$ examples are retrieved from $\mathcal{X}$ stored in the vector database. $\mathcal{M}$ is then used as context for the LLM to provide the most relevant examples for annotating the input text $\mathcal{T}_i$.

3.2.5 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[2]. To solve the token-label misalignment problem, in this study, we leverage 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`.

4 Experimental Setup

4.1 Datasets

In this study, 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 of Appendix A. 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.

CoNLL-2003 The CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) dataset consists of four general entity types. 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. This dataset is particularly challenging due to its noisy text, sparse entity occurrences, and limited labeled examples per category. Improving recall on this dataset remains a significant challenge for LLMs.

GUM The GUM (Zeldes, 2017) dataset is a richly annotated corpus designed for multiple NLP tasks, including NER. It captures linguistic phenomena across various domains and genres, making it a valuable resource for evaluating model performance. The dataset includes eleven distinct named entity types. Compared to CoNLL-2003 and WNUT-17, GUM presents a higher level of complexity by incorporating a diverse set of entity types spanning multiple domains.

SKILLSPAN The SKILLSPAN (Zhang et al., 2022a) dataset is composed of a single entity type. Unlike traditional entities, soft skills do not follow a fixed syntactic or semantic structure, making them inherently ambiguous. These entities can range from single tokens to multi-token expressions, increasing the complexity of annotation and information extraction tasks for LLMs.

4.2 Approaches 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. This zero-shot setting allows the evaluation of the LLM's general knowledge of the task. Moreover, ICL and RAG-based approaches, detailed in Section 3.2.3 and Section 3.2.4 respectively, are considered. For both, experiments are conducted with three different numbers of context examples: *(i)* 25, *(ii)* 50, and *(iii)* 75. Experiments are conducted on a 30% sample of the training set $\mathcal{D}_{\text{train}}$, while the ablation study in Appendix D examines the effects of 10% and 20% sample sizes.

This paper considers five different LLMs[3]: *(i)* `gpt-4o-mini-2024-07-18`, *(ii)* `Qwen2.5-72B-Instruct`, *(iii)* `Llama3.5-70B-Instruct`, *(iv)* `Qwen2.5-7B-Instruct`, and *(v)* `Llama3.1-8B-Instruct`, and two embeddings models: *(i)* the `text-embedding-3-large` model[4], and *(ii)* the `sentence transformer all-MiniLM-L6-v2` model (Reimers and Gurevych, 2019).

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

[3] The models are referred to by their base names, such as `Qwen2.5-72B` for `Qwen2.5-72B-Instruct`, and so on.

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

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. Implementation details of results are reported Appendix B.

4.3 NER Evaluation Process

To assess the quality of annotations generated by LLMs, the RoBERTa model (Liu et al., 2019) is fine-tuned on LLM-annotated datasets, leveraging its proven effectiveness in NER tasks (Zhou et al., 2022; Zhang et al., 2022a). Initially, an LLM is employed to automatically annotate sentences in $\mathcal{T} \subset \mathcal{D}_{train}$, using strategies from Section 3.2. This process generates annotations for $\mathcal{T}$, resulting in a new training set, $\hat{\mathcal{T}}$, with $|\hat{\mathcal{T}}| = |\mathcal{T}|$. This annotated set is then used to fine-tune the RoBERTa model (Liu et al., 2019). Model selection is performed on the validation set, $\mathcal{D}_{valid}$, and the final evaluation results are based on the test set, $\mathcal{D}_{test}$. To ensure robustness and mitigate the impact of random initialization, we average the results across five different seed values. The F_1 score is used to assess the performances of the models.

5 Results and Analysis

This section presents the quantitative results of this study, as well as its analysis. Qualitative results are reported in Appendix F, while Appendix E reports the statistical tests to support the findings.

5.1 Quantitative Results

Figure 2 presents the overall results of the experiments, while the corresponding detailed outcomes are reported in Appendix C. Specifically, the heatmaps present the F_1 scores obtained on the test set for different datasets, comparing several models and methods used in the proposed 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 (also shows statistical significance over other models as detailed in Appendix E), achieving an F_1 score of 89.72 with 75 context examples, which is just 2.7% below human-level annotation. Among the $\sim$ $70B$ models, `Qwen2.5-72B` with OpenAI embeddings performs comparably to `gpt-4o-mini` with an F_1 score of 89.34, while `Llama3.5-70B` with ST embeddings lags slightly behind with an F_1 score of 87.33. At the $\sim$ $7B$ scale, `Qwen2.5-7B` with ST embeddings significantly outperforms its counterparts, achieving an F_1 score of 87.94, while `Llama3.1-8B` with OpenAI embeddings scores 84.91. 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 significantly from larger context sizes of 75 examples, while `Llama3.5-70B` performs best at a slightly lower context size. This suggests that different models have varying levels of context saturation, 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, `Qwen2.5-70B` with OpenAI embeddings achieves the highest F_1 score of 53.72, slightly outperforming `gpt-4o-mini`, which attains an F_1 score of 53.43. The `Llama3.5-70B` model exhibits inconsistent performance, scoring 51.18 with ICL at 75 context examples, suggesting that it struggles to generalize well for rare entity detection. At the $\sim$ $7B$ scale, `Qwen2.5-7B` with ST embeddings achieves an F_1 score of 49.48, significantly outperforming `Llama3.1-8B`, which scores 44.42. 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 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 strong embeddings.

The GUM dataset presents a unique challenge due to its diverse entity types, requiring models to generalize across various linguistic structures. `Qwen2.5-70B` with ST embeddings achieves the best F_1 score of 55.11, significantly surpassing `gpt-4o-mini`, which attains an F_1 score of 52.28, and `Llama3.5-70B` with OpenAI embeddings, which achieves an F_1 score of 48.33. At the $\sim$ $7B$ scale, `Qwen2.5-7B` with OpenAI embeddings

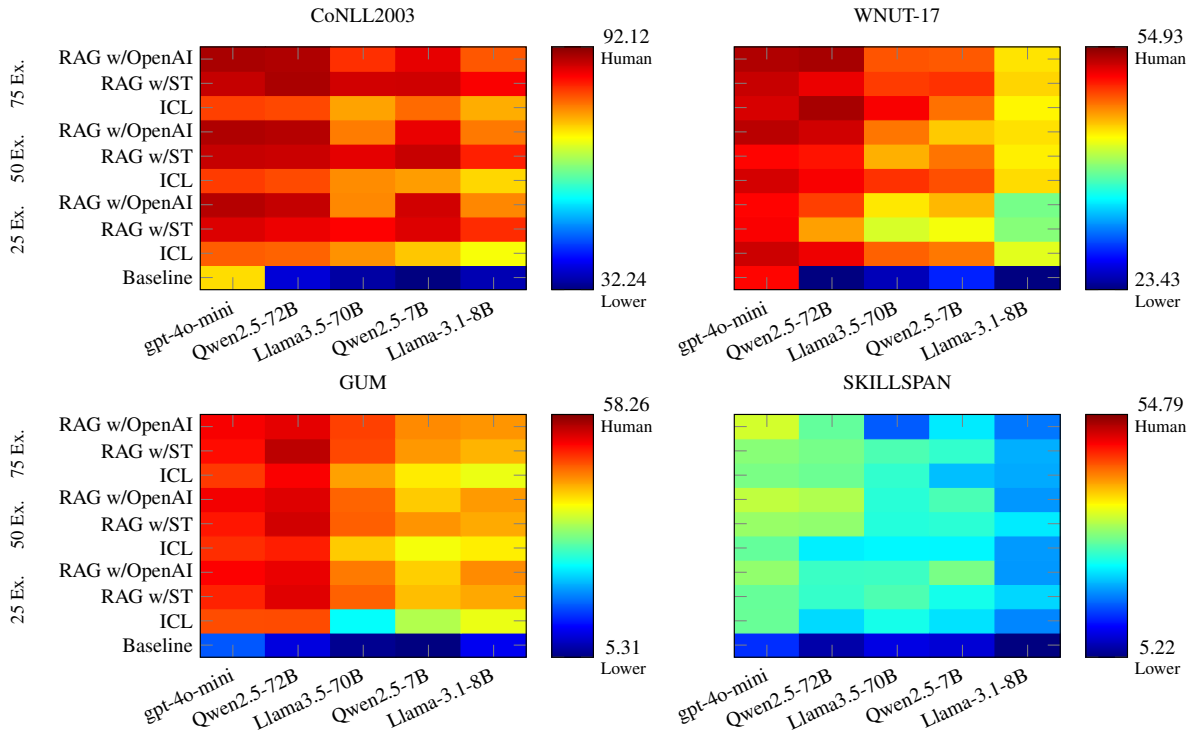

Figure 2: Heatmaps of the F_1 scores across four datasets. The color scale represents performance, with red indicating higher scores reaching human-level, and blue indicating lower scores starting from the lowest performing model

achieves an F_1 score of 44.48, outperforming `Llama3.1-8B`, which scores 43.91. However, both models show a notable performance drop compared to their larger counterparts, suggesting that smaller models struggle with datasets with diverse entities. The 3.15% gap between the best-performing LLM and human-level annotation highlights that GUM remains a challenging dataset for LLMs. The heatmap further suggests that model performance fluctuates significantly depending on context size and embedding choice.

The SKILLSPAN dataset is the most difficult among those evaluated, as it requires understanding nuanced skill mentions across various job contexts. `gpt-4o-mini` with OpenAI embeddings performs the best, achieving an F_1 score of 34.06 with 75 context examples, but this is still far from human-level annotation. At the $\sim$ $70B$ scale, `Qwen2.5-70B` with ST embeddings achieves an F_1 score of 32.35 with 50 context examples, outperforming `Llama3.5-70B`, which achieves an F_1 score of 27.55. Among $\sim$ $7B$ models, `Qwen2.5-7B` with OpenAI embeddings achieves an F_1 score of 29.67, significantly surpassing `Llama3.1-8B`, which scores 22.88. 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.

5.2 Different Sample Space Choices

This section examines the impact of sample space choices, denoted as $\mathcal{X}$ in Section 3.1, using the proposed RAG-based approach as overall it performs better than ICL. The experiments are conducted on the SKILLSPAN dataset with the `gpt-4o-mini` model and OpenAI embeddings. As shown in Figure 3, for smaller dataset splits, the RAG-based approach exhibits greater variability, similar to the behavior seen with ICL. This suggests that as the sample space for selecting context examples decreases, the performance of the RAG-based approach converges more closely with that of ICL. More detailed results are reported in Appendix D.

6 Discussion

Performance of LLMs The performance of different LLMs in our study reveals interesting

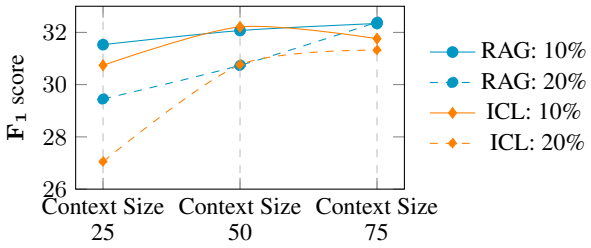

Figure 3: F_1 scores for different context sizes (25, 50, and 75) and sample spaces (10% and 20%) for the RAG and ICL approach on the SKILLSPAN dataset, using the `gpt-4o-mini` model. The plot indicates that with a smaller sample size, the RAG approach performs comparably to ICL.

insights. Across all datasets, 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, particularly SKILLSPAN. While all models perform well on CoNLL-2003, performance declines as entity structures become more complex, such as in GUM and SKILLSPAN.

Effect of Embeddings The choice of embeddings for retrieval of context for LLMs plays a crucial role in annotation quality in retrieval-based methods. OpenAI embeddings lead to better F_1 scores compared to smaller-scale ST embeddings especially for `gpt-4o-mini` model. This effect is particularly evident in WNUT-17 and GUM, where entity distributions are more diverse, and high-quality embeddings improve retrieval effectiveness. In contrast, SKILLSPAN remains challenging across all embedding strategies, suggesting that current embedding techniques struggle with soft skill representation due to the abstract nature of the entities.

Effect of Model Size Larger models generally perform better, but retrieval quality is equally critical. `Qwen2.5-7B` slightly outperforms `Llama3.1-8B` and performs comparably to `Llama3.5-70B` with proper embeddings, indicating that architecture and training data impact annotation beyond parameter count. Statistical tests in Appendix E support this finding.

Effect of Dataset Complexity Breaking down results per dataset, CoNLL-2003 shows minimal variance across methods, as structured entities are well-represented in training data. WNUT-17 benefits the most from retrieval-based methods, as rare entities require additional context for accurate recognition. GUM's diverse entity types pose a challenge for ICL, but RAG-based methods significantly improve performance. Finally, SKILLSPAN remains the most difficult dataset, with lower performance across all methods, underscoring the limitations of LLMs and embeddings in capturing the semantics of soft skills.

7 Conclusions and Future Works

This study systematically evaluates the effectiveness of LLMs for data annotation across four diverse datasets—CoNLL-2003, WNUT-17, GUM, and SKILLSPAN of varying complexity. It compares RAG in different embedding strategies, ICL, and a baseline approach. The results demonstrate that RAG-based methods consistently outperform both ICL and the baseline across all datasets, significantly reducing the performance gap with human-level annotation.

A key finding is that dataset complexity plays a crucial role in model performance. For structured datasets like CoNLL-2003, LLMs perform exceptionally well, with models such as `gpt-4o-mini` and `Qwen2.5-72B` achieving results within 3% of human-level annotation. Conversely, performance deteriorates as dataset complexity increases. The SKILLSPAN dataset, which requires nuanced skill recognition, presents the greatest challenge, with LLMs struggling to capture implicit skill mentions.

Our analysis also highlights the importance of context size and embedding choice in retrieval-augmented annotation. We observe that larger models such as `Qwen2.5-72B` and `gpt-4o-mini` benefit from larger context sizes, while smaller models like `Qwen2.5-7B` can still perform competitively when paired with high-quality sentence embeddings. However, models exhibit context saturation effects, where additional examples do not always lead to linear performance improvements.

Future works will focus on enhancing the performance of LLMs for complex datasets, particularly in specialized domains. In addition, future works will expand the study to more LLMs and to different NLP tasks.

Limitations

In this study, we evaluate LLMs for data annotation tasks and introduce a RAG-based approach with different embedding models to enhance performance on NER datasets. However, our work has several limitations that highlight areas for future research.

First, our 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 generalizability.

Second, for the proof of concept, we employ a naïve RAG approach for context selection. Future work could explore more sophisticated retrieval techniques, such as adaptive retrieval strategies, re-ranking mechanisms, or hybrid approaches combining dense and sparse retrieval, to further optimize performance.

Third, our 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 diverse and underrepresented datasets, could provide valuable insights.

References

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data. *Preprint*, arXiv:2402.15343.

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. *Preprint*, arXiv:2005.14165.

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

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.

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. *Preprint*, arXiv:2301.00234.

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. *Preprint*, arXiv:2406.12334.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. 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.

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. *Preprint*, arXiv:2312.02296.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024a. Annollm: Making large language models to be better crowdsourced annotators. *Preprint*, arXiv:2303.16854.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024b. 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.

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

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.

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

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.

Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024. A simple but effective approach to improve structured language model output for information extraction. *Preprint*, arXiv:2402.13364.

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.

Yuji Naraki, Ryosuke Yamaki, Yoshikazu Ikeda, Takafumi Horie, and Hiroki Naganuma. 2024. Augmenting ner datasets with llms: Towards automated and refined annotation. *Preprint*, arXiv:2404.01334.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

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. *Preprint*, arXiv:2312.11805.

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. *Preprint*, arXiv:2302.13971.

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. *Preprint*, arXiv:2402.14558.

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. *Preprint*, arXiv:2304.10428.

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.

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. *Preprint*, arXiv:2201.11903.

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. *Preprint*, arXiv:2310.01469.

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

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*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *Preprint*, arXiv:2210.03493.

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

A Datasets Statistics

Table 1: Statistics of the datasets considered in this study. The average entity length refers to the average number of tokens for each entity.

Dataset	Sentences			Tokens			Avg. Entity Length
	Train	Validation	Test	Train	Validation	Test	
CoNLL-2003	14041	3250	3453	203621	51362	46435	1.60
WNUT-2017	3394	1008	1287	62730	15734	23394	1.73
GUM	1435	615	805	29392	12688	17437	3.15
SKILLSPAN	3074	1396	1522	92621	39923	42541	4.72

Table 1 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.

Moreover, we discuss below 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 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 instructed fine-tuned variants of the open-source models are employed in the proposed study. The models are used only for inference, with 4-bit quantization (Jacob et al., 2018). The experiments with billion scale models are conducted on an $A100$ GPU with a seed value of 42. All experiments to fine-tune NER task are performed with the RoBERTa model, available via HuggingFace (Hugging Face, 2023), are conducted in a python environment, on an RTX $A5000$ GPU. The experiments are performed using the following five seed values: $[23112, 13215, 6465, 42, 5634]$. Moreover, the statistical significance tests are performed with the help of scikit-posthocs (Terpilowski, 2019) library available in python.

C Complete Results

Table 2: The F_1, precision and recall along with standard deviation are reported on the test set. The values are averaged over five different random initializations. #Ex. represents the number of context examples used. Baseline refers to the use of LLM with no context examples.

#Ex.	Method	CoNLL2003			WNUT-17			GUM			SKILLSPAN		
		P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**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	$87.43_{\pm 0.48}$	$91.39_{\pm 0.16}$	$\mathbf{89.36}_{\pm 0.27}$	$56.53_{\pm 2.35}$	$50.29_{\pm 2.64}$	$\mathbf{53.14}_{\pm 0.75}$	$47.32_{\pm 0.92}$	$58.44_{\pm 1.21}$	$\mathbf{52.28}_{\pm 0.65}$	$23.88_{\pm 1.09}$	$54.28_{\pm 2.26}$	$\mathbf{33.13}_{\pm \mathbf{0.77}}$
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	$88.07_{\pm 0.35}$	$91.44_{\pm 0.28}$	$\mathbf{89.72}_{\pm 0.25}$	$55.72_{\pm 4.22}$	$51.71_{\pm 3.34}$	$\mathbf{53.43}_{\pm 0.54}$	$47.04_{\pm 1.29}$	$58.19_{\pm 1.18}$	$\mathbf{52.02}_{\pm 1.15}$	$24.66_{\pm 1.34}$	$55.39_{\pm 3.19}$	$\mathbf{34.06}_{\pm 0.88}$

Table 3: The F_1, precision and recall along with standard deviation are reported on the test set. The values are averaged over five different random initializations. #Ex. represents the number of context examples used. Baseline refers to the use of LLM with no context examples.

#Ex.	Method	CoNLL2003			WNUT-17			GUM			SKILLSPAN		
		P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**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	$86.77_{\pm 0.54}$	$92.05_{\pm 0.72}$	$\mathbf{89.34}_{\pm 0.61}$	$48.56_{\pm 2.08}$	$60.22_{\pm 1.52}$	$\mathbf{53.72}_{\pm 0.71}$	$47.24_{\pm 1.27}$	$60.34_{\pm 0.57}$	$52.98_{\pm 0.76}$	$19.95_{\pm 0.74}$	$50.74_{\pm 1.47}$	$28.62_{\pm 0.77}$
	Llama3.5-70B-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	$83.15_{\pm 1.42}$	$86.37_{\pm 0.90}$	$\mathbf{84.72}_{\pm 0.54}$	$36.68_{\pm 1.32}$	$49.10_{\pm 3.77}$	$41.89_{\pm 0.99}$	$43.09_{\pm 1.10}$	$50.88_{\pm 2.31}$	$\mathbf{46.63}_{\pm 0.89}$	$\mathbf{1}9.62_{\pm 1.44}$	$46.47_{\pm 1.76}$	$\mathbf{27.55}_{\pm 1.21}$
	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 4: The F_1, precision and recall along with standard deviation are reported on the test set. The values are averaged over five different random initializations. #Ex. represents the number of context examples used. Baseline refers to the use of LLM with no context examples.

#Ex.	Method	CoNLL2003			WNUT-17			GUM			SKILLSPAN		
		P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**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 Further Results on Different Sample Space Choices

Tables 5 examine the influence of sample space $\mathcal{X}$ and context size $\mathcal{M}$ 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 of 32.39% at 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. These findings underscore the importance of context size and external knowledge availability in optimizing RAG-based methods.

Table 5: Study comparing RAG and ICL methods at different size of sample spaces (10% and 20%) and context sizes (25, 50, and 75). Experiments were conducted on the SKILLSPAN dataset using the gpt-4o-mini-2024-07-18 model. The results are presented with standard deviations, showing how performance metrics vary across sampling choices and context sizes for both methods.

Sample Space	Context Size	Precision	Recall	F1 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	$22.89_{\pm 1.11}$	$49.78_{\pm 2.89}$	$\mathbf{31.32}_{\pm 0.99}$

E 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, we conducted a non-parametric test. This test helps us 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 $\alpha = 0.01$, we reject the null hypothesis, 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

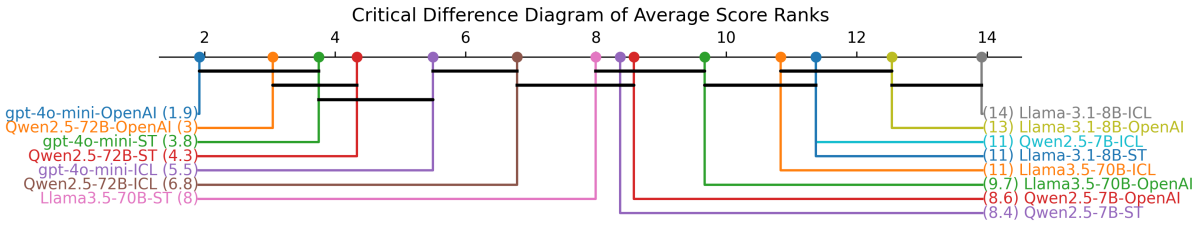

Figure 4: Critical Difference diagram of average score ranks. The models connected with horizontal line shows no statistical difference. The models with lower ranks shows superior performance than those of higher ranks.

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 114.42 with a p-value of 7.71^{-18}, we reject the null hypothesis, suggesting that at least one model shows a statistically significant difference in performance. Consequently, we conducted the post-hoc Conover test. Figure 4 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 1.9, indicating it consistently outperformed other approaches. Other strong performers include `Qwen2.5-72B-OpenAI` (3), `gpt-4o-mini-ST` (3.8), and `Qwen2.5-72B-ST` (4.3). These models have lower rankings and are clustered towards the left. In contrast, `Llama3.1-8B-ICL` (14), `Llama3.1-8B-OpenAI` (13), and `Qwen2.5-7B-ICL` (11) 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 `Llama3.5-70B`, 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 `Llama3.5-70B` using ST embeddings and `Qwen2.5-72B` using ICL. 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.

F Qualitative Analysis

This study broadly explores the efficacy of LLMs for data annotation tasks. Four different datasets of varying complexity are chosen. From Table 2, it is observed 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 (more details are reported in Section 4.1), 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, as indicated by Tables 2, 3, and 4, 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 6 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

Table 6: 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.

information from the context examples when the task is difficult. From Tables 2, 3, and 4, it is observed that LLM-generated annotations improve recall, whereas precision is compromised. Table 6 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.

G Prompt

This section presents the prompts used to generate the response 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.

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}

H Examples

This section provides examples of prompts from the training data for different datasets used in this study. For visual purposes, we used only only $top5$ examples in context. Follows several prompt examples for the: *(i)* CoNLL-2003, *(ii)* WNUT-17, *(iii)* SKILLSPAN datasets, and *(iv)* GUM datasets.

Example 1–CoNLL-2003

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:

['PER', 'ORG', 'LOC', 'MISC']

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

```
["A South African boy is writing back to an American girl whose message in a
    bottle he found washed up on President Nelson Mandela 's old prison island
     .", [{'Entity': 'South African', 'Label': 'MISC'}, {'Entity': 'American',
```

```
    'Label': 'MISC'}, {'Entity': 'Nelson Mandela', 'Label': 'PER'}]]

['A rottweiler dog belonging to an elderly South African couple savaged to
    death their two-year-old grandson who was visiting , police said on
    Thursday .', [{'Entity': 'South African', 'Label': 'MISC'}]]

['The princess , who has carved out a major role for herself as a helper of
    the sick and needy , is said to have turned to Mother Teresa for guidance
    as her marriage crumbled to heir to the British throne Prince Charles .',
    [{'Entity': 'Mother Teresa', 'Label': 'PER'}, {'Entity': 'British', 'Label
    ': 'MISC'}, {'Entity': 'Prince Charles', 'Label': 'PER'}]]

['South African answers U.S. message in a bottle .', [{'Entity': 'South
    African', 'Label': 'MISC'}, {'Entity': 'U.S.', 'Label': 'LOC'}]]

["But Carlo Hoffmann , an 11-year-old jailer 's son who found the bottle on
    the beach at Robben Island off Cape Town after winter storms , will send
    his letter back by ordinary mail on Thursday , the post office said .",
    [{'Entity': 'Carlo Hoffmann', 'Label': 'PER'}, {'Entity': 'Robben Island',
     'Label': 'LOC'}, {'Entity': 'Cape Town', 'Label': 'LOC'}]]
```

Input

Revered skull of S. Africa king is Scottish woman 's .

Response

[Entity: S. Africa, Label: LOC, Entity: Scottish, Label: MISC]

Example 2–CoNLL-2003

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:

['PER', 'ORG', 'LOC', 'MISC']

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

```
['Rwanda said on Saturday that Zaire had expelled 28 Rwandan Hutu refugees
    accused of being " trouble-makers " in camps in eastern Zaire .', [{'
    Entity': 'Rwanda', 'Label': 'LOC'}, {'Entity': 'Zaire', 'Label': 'LOC'},
    {'Entity': 'Rwandan', 'Label': 'MISC'}, {'Entity': 'Hutu', 'Label': 'MISC
    '}, {'Entity': 'Zaire', 'Label': 'LOC'}]]

['Repatriation of 1.1 million Rwandan Hutu refugees announced by Zaire and
    Rwanda on Thursday could start within the next few days , an exiled
    Rwandan Hutu lobby group said on Friday .', [{'Entity': 'Rwandan Hutu', '
    Label': 'MISC'}, {'Entity': 'Zaire', 'Label': 'LOC'}, {'Entity': 'Rwanda',
     'Label': 'LOC'}, {'Entity': 'Rwandan Hutu', 'Label': 'MISC'}]]

['Innocent Butare , executive secretary of the Rally for the Return of
    Refugees and Democracy in Rwanda ( RDR ) which says it has the support of
    Rwanda \'s exiled Hutus , appealed to the international community to deter
     the two countries from going ahead with what it termed a " forced and
    inhuman action " .', [{'Entity': 'Innocent Butare', 'Label': 'PER'}, {'
    Entity': 'Rally for the Return of Refugees and Democracy in Rwanda', '
    Label': 'ORG'}, {'Entity': 'RDR', 'Label': 'ORG'}, {'Entity': 'Rwanda', '
    Label': 'LOC'}, {'Entity': 'Hutus', 'Label': 'MISC'}]]
```

```
['Rwanda says Zaire expels 28 Rwandan refugees .', [{'Entity': 'Rwanda', '
    Label': 'LOC'}, {'Entity': 'Zaire', 'Label': 'LOC'}, {'Entity': 'Rwandan',
     'Label': 'MISC'}]]

['Rwandan group says expulsion could be imminent .', [{'Entity': 'Rwandan', '
    Label': 'MISC'}]]
```

Input

Captain Firmin Gatera , spokesman for the Tutsi-dominated Rwandan army , told Reuters in Kigali that 17 of the 28 refugees handed over on Friday from the Zairean town of Goma had been soldiers in the former Hutu army which fled to Zaire in 1994 after being defeated by Tutsi forces in Rwanda 's civil war .

Response

[Entity: Captain Firmin Gatera, Label: PER, Entity: Rwandan, Label: MISC, Entity: Reuters, Label: ORG, Entity: Kigali, Label: LOC, Entity: Zairean, Label: MISC, Entity: Goma, Label: LOC, Entity: Hutu, Label: MISC, Entity: Zaire, Label: LOC, Entity: Tutsi, Label: MISC, Entity: Rwanda, Label: LOC]

Example 3–WNUT-17

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:

['corporation', 'creative-work', 'group', 'location', 'person', 'product']

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

```
['@justinbieber i just wanna say you make me smile everyday :) thanks for
    smiling because when u smile i smile ! :)', []]

["@joeymcintyre I heart you . Even if I haven't seen u in months ... SEND A
    PIC !", []]

['@lovable_sin OMG OMG OMG ! Thank you for " tumblring " it to me , I so wasn
    \'t expecting them today . OMG !', []]

['RT @aplusk : This made me laugh today http://bit.ly/bjOhom &lt; --- courtesy
     of splurb . What made you laugh ?', []]

['RT @Sn00ki : Haha yes !!! I love that you knew that :) RT @trishamelissa
    @Sn00ki Is phenomenal the word of the day ?', []]
```

Input

@jimmyfallon is following me ! OMG ! My life is now complete ! I heart you JF and have for years ! Thank you for making me laugh everyday !

Response

[Entity: @jimmyfallon, Label: person, Entity: JF, Label: person]

Example 4–WNUT-17

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:

['corporation', 'creative-work', 'group', 'location', 'person', 'product']

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

```
['We are one step closer to our new kitchens . We chose a maker and had
    official measurements taken today !', []]

['We were all enjoying a glass of wine in the office when a fudge delivery
    showed up . I love my job . And I love Fridays .', []]

['800 miles to see clients , 3 ACC candidate/commissioner meetings , big press
     release , making it to Friday .. PRICELESS !', []]

["I hope the weeks keep flying . It 's actually fantastic the way none of the
    days dragged this week .... like NONE . :D", []]

['Feeling really good after great week in our SF and LA offices . Glad to kick
     back on AMerican flight back to NYC', [{'Entity': 'SF', 'Label': '
    location'}, {'Entity': 'LA', 'Label': 'location'}, {'Entity': 'AMerican',
    'Label': 'corporation'}, {'Entity': 'NYC', 'Label': 'location'}]]
```

Input

Great week in the Optimise office, another new client on board and we are close to signing a new team member

---

Response

[Entity: Optimise, Label: corporation]

Example 5–SKILLSPAN

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:

['Skill']

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

```
['Hands on experience with automated testing using Java .', []]

['Experience with automation systems framework design/use and deployment .',
    []]

['Good understanding of Agile methodologies and Continuous Delivery .', []]

['Demonstrate clear understanding of automation and orchestration principles
    .', []]

['Good exposure to UI Frameworks like Angular Proficiency in SQL and Database
    development .', []]
```

```
["Ability to understand and use efficient Defect management regular view of
    test coverage to identify gaps and provide improvements Personal
    Specification 5+ years of relevant IT/quality assurance work experience
    Bachelor's degree in Computer Science or related field of study or
    equivalent relevant experience; demonstrated experience within the quality
     assurance / testing arena; demonstrated skills in quality assurance
    methods/processes and practices .", [{'Entity': 'understand and use
    efficient Defect management', 'Label': 'Skill'}, {'Entity': 'identify gaps
    ', 'Label': 'Skill'}]]
```

Input

Very good understanding of test automation frameworks.

Response

[Entity: test automation frameworks, Label: Skill]

Example 6–SKILLSPAN

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:

['Skill']

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

```
['Strong communication skills including the ability to express complex
    technical concepts to different audiences in writing and conference calls
    .', [{'Entity': 'communication skills', 'Label': 'Skill'}, {'Entity': '
    express complex technical concepts to different audiences', 'Label': '
    Skill'}]]

['Excellent organizational verbal and written communication skills .', [{'
    Entity': 'organizational verbal and written communication skills', 'Label
    ': 'Skill'}]]

['Excellent organizational verbal and written communication skills .', [{'
    Entity': 'organizational verbal and written communication skills', 'Label
    ': 'Skill'}]]

['The ability to work within a team and in collaboration with others is
    critical to this position and excellent communication skills verbal and
    written are essential .', [{'Entity': 'work within a team and in
    collaboration with others', 'Label': 'Skill'}, {'Entity': 'communication
    skills', 'Label': 'Skill'}]]

['This role requires a wide variety of strengths and capabilities including
    Ability to work collaboratively in teams and develop meaningful
    relationships to achieve common goals Strong organizational skills Ability
     to multi-task and deliver to a tight deadline Excellent written and
    verbal communication skills Experience developing UI components in Angular
     Good experience in using design patterns UML OO concepts .', [{'Entity':
    'work collaboratively in teams', 'Label': 'Skill'}, {'Entity': 'develop
    meaningful relationships', 'Label': 'Skill'}, {'Entity': 'achieve common
    goals', 'Label': 'Skill'}, {'Entity': 'organizational skills', 'Label': '
    Skill'}, {'Entity': 'multi-task', 'Label': 'Skill'}, {'Entity': 'deliver
    to a tight deadline', 'Label': 'Skill'}, {'Entity': 'communication skills
    ', 'Label': 'Skill'}, {'Entity': 'developing UI components', 'Label': '
```

```
Skill'}, {'Entity': 'using design patterns', 'Label': 'Skill'}]]
```

Input

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 .

Response

[Entity: verbal and written skills, Label: Skill, Entity: communicate effectively on both a technical and business level, Label: Skill, Entity: work under pressure, Label: Skill, Entity: resolve issues affecting the production services, Label: Skill]

Example 7–GUM

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:

['abstract', 'animal', 'event', 'object', 'organization', 'person', 'place', 'plant', 'quantity', 'substance', 'time']

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

```
[ 'The 131-page document was found on Castlefrank Road in Kanata , Ontario in
    a rain-stained , tire-marked brown envelope by a passerby', 'Entities':
    [{'Entity': 'The 131-page document was found on Castlefrank Road in Kanata
     , Ontario in a rain-stained , tire-marked brown envelope by a passerby',
    'Label': 'event'}]]

['Also the language is important in writing and in literature', 'Entities':
    [{'Entity': 'the language', 'Label': 'abstract'}, {'Entity': 'writing', '
    Label': 'abstract'}, {'Entity': 'literature', 'Label': 'abstract'}]]

['Ingredients Basil comes in many different varieties , each of which have a
    unique flavor and smell', 'Entities': [{'Entity': 'Ingredients', 'Label':
    'object'}, {'Entity': 'Basil', 'Label': 'plant'}, {'Entity': 'many
    different varieties', 'Label': 'abstract'}, {'Entity': 'each of which', '
    Label': 'abstract'}, {'Entity': 'a unique flavor and smell', 'Label': '
    abstract'}]]

['We do not want to just traffic in the same 24 hour news cycle', 'Entities':
    [{'Entity': 'We do not want to just traffic in the same 24 hour news cycle
    ', 'Label': 'abstract'}]]

['You go through quite a bit', 'Entities': [{'Entity': 'You', 'Label': 'person
    '}, {'Entity': 'quite a bit', 'Label': 'quantity'}]]
```

Input

If you are just visiting York for the day , using a Park and Ride [1] costs a lot less than trying to park in or near the city centre , and there are five sites dotted around the Outer Ring Road

---

Response

['Entity': 'York', 'Label': 'place', 'Entity': 'the day', 'Label': 'time', 'Entity': 'a Park and Ride', 'Label': 'object', 'Entity': 'the city centre', 'Label': 'place', 'Entity': 'five sites', 'Label': 'quantity', 'Entity': 'the Outer Ring Road', 'Label': 'place']

Example 8–GUM

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:

['abstract', 'animal', 'event', 'object', 'organization', 'person', 'place', 'plant', 'quantity', 'substance', 'time']

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

```
['" NASA Administrator Charles Bolden announces where four space shuttle
    orbiters will be permanently displayed at the conclusion of the Space
    Shuttle Program during an event commemorating the 30th anniversay of the
    first shuttle launch on April 12 , 2011', 'Entities': [{'Entity': 'NASA
    Administrator Charles Bolden', 'Label': 'person'}, {'Entity': 'where four
    space shuttle orbiters will be permanently displayed', 'Label': 'place'},
    {'Entity': 'the conclusion of the Space Shuttle Program', 'Label': 'event
    '}, {'Entity': 'an event', 'Label': 'event'}, {'Entity': '30th anniversay
    of the first shuttle launch', 'Label': 'event'}, {'Entity': 'April 12 ,
    2011', 'Label': 'time'}]]

['NASA celebrated the launch of the first space shuttle Tuesday at an event at
     the Kennedy Space Center ( KSC ) in Cape Canaveral , Florida', 'Entities
    ': [{'Entity': 'NASA', 'Label': 'organization'}, {'Entity': 'the launch of
     the first space shuttle', 'Label': 'event'}, {'Entity': 'Tuesday', 'Label
    ': 'time'}, {'Entity': 'an event', 'Label': 'event'}, {'Entity': 'Kennedy
    Space Center', 'Label': 'place'}, {'Entity': 'KSC', 'Label': 'place'}, {'
    Entity': 'Cape Canaveral , Florida', 'Label': 'place'}]]

['Looking back : Space Shuttle Columbia lifts off on STS-1 from Launch Pad 39A
     at the Kennedy Space Center on April 12 , 1981', 'Entities': [{'Entity':
    'Space Shuttle Columbia', 'Label': 'object'}, {'Entity': 'STS-1', 'Label':
     'event'}, {'Entity': 'Launch Pad 39A', 'Label': 'place'}, {'Entity': '
    Kennedy Space Center', 'Label': 'place'}, {'Entity': 'April 12 , 1981', '
    Label': 'time'}]]

['At the ceremony , NASA Administrator Charles Bolden announced the locations
    that would be given the three remaining Space Shuttle orbiters following
    the end of the Space Shuttle program', 'Entities': [{'Entity': 'the
    ceremony', 'Label': 'event'}, {'Entity': 'NASA Administrator Charles
    Bolden', 'Label': 'person'}, {'Entity': 'the locations', 'Label': 'place
    '}, {'Entity': 'the three remaining Space Shuttle orbiters', 'Label': '
    object'}, {'Entity': 'the end of the Space Shuttle program', 'Label': '
    event'}]]

['On April 12 , 1981 , Space Shuttle Columbia lifted off from the Kennedy
    Space Center on STS-1 , the first space shuttle mission', 'Entities': [{'
    Entity': 'April 12 , 1981', 'Label': 'time'}, {'Entity': 'Space Shuttle
    Columbia', 'Label': 'object'}, {'Entity': 'Kennedy Space Center', 'Label':
     'place'}, {'Entity': 'STS-1', 'Label': 'event'}, {'Entity': 'the first
    space shuttle mission', 'Label': 'event'}]]
```

Input

Tuesday , September 22 , 2015 Discovery is undergoing decommissioning and currently being prepped for display by removing toxic materials from the orbiter

---

Response

['Entity': 'Tuesday', 'Label': 'time', 'Entity': 'September 22 , 2015', 'Label': 'time', 'Entity': 'Discovery', 'Label': 'object', 'Entity': 'decommissioning', 'Label': 'event', 'Entity': 'display', 'Label': 'event', 'Entity': 'toxic materials',

’Label’: ’substance’, ’Entity’: ’the orbiter’, ’Label’: ’object’]