LLMs are Better Than You Think: Label-Guided In-Context Learning for Named Entity Recognition

Fan Bai[♣] **Hamid Hassanzadeh**[♢] **Ardavan Saeedi**[♢] **Mark Dredze**[♣]
[♣]Johns Hopkins University, [♢]Optum
`fbai3@jh.edu`, `{hamid.hassanzadeh, ardavan.saeedi}@optum.com`, `mdredze@cs.jhu.edu`

Abstract

In-context learning (ICL) enables large language models (LLMs) to perform new tasks using only a few demonstrations. However, in Named Entity Recognition (NER), existing ICL methods typically rely on task-agnostic semantic similarity for demonstration retrieval, which often yields less relevant examples and leads to inferior results. We introduce **DEER**, a training-free ICL approach that enables LLMs to make more informed entity predictions through the use of label-grounded statistics. DEER leverages token-level statistics from training labels to identify tokens most informative for entity recognition, enabling entity-focused demonstrations. It further uses these statistics to detect and refine error-prone tokens through a targeted reflection step. Evaluated on five NER datasets across four LLMs, DEER consistently outperforms existing ICL methods and achieves performance comparable to supervised fine-tuning. Further analyses demonstrate that DEER improves example retrieval, remains effective on both seen and unseen entities, and exhibits strong robustness in low-resource settings.[1]

1 Introduction

Identifying and categorizing named entities in text (Named Entity Recognition, NER) is a core information extraction (IE) task with applications in diverse domains including science (Lin et al., 2004; Leser and Hakenberg, 2005; Luan et al., 2018; Bai et al., 2022), news (Sang and De Meulder, 2003; Whitelaw et al., 2008), and social media (Finin et al., 2010; Ritter et al., 2011; Li et al., 2012). Supervised learning works well for NER using models with tailored architectures, fine-tuned on large labeled datasets (Nadeau and Sekine, 2007; Ma and Hovy, 2016). However, these approaches do not generalize well across domains or to new entity types without re-training on annotated data, thereby constraining their broader applicability (Lin and Lu, 2018; Bai et al., 2021).

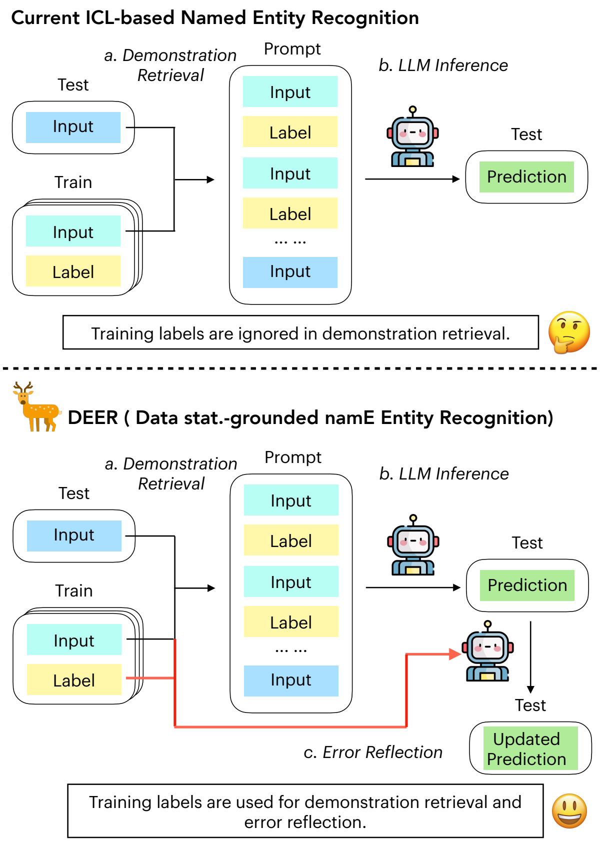
Current ICL-based Named Entity Recognition
a. Demonstration Retrieval
Prompt
b. LLM Inference
Test
Input
Train
Input
Label
Input
Label
Input
Label
… …
Input
Test
Prediction
Training labels are ignored in demonstration retrieval.
DEER (Data stat.-grounded namE Entity Recognition)
a. Demonstration Retrieval
Prompt
b. LLM Inference
Test
Input
Train
Input
Label
Input
Label
Input
Label
… …
Input
Test
Prediction
Test
Updated Prediction
c. Error Reflection
Training labels are used for demonstration retrieval and error reflection.

Figure 1: Existing ICL methods for NER typically select demonstrations based on task-agnostic input embeddings, which we argue yield suboptimal demonstrations for LLMs and consequently inferior performance. In contrast, our proposed method, DEER, leverages label-grounded statistics to identify entity-relevant tokens, enabling more task-focused demonstration selection and targeted error reflection.

Recent advances in large language models (LLMs; Brown et al., 2020; Touvron et al., 2023; Dubey et al., 2024; OpenAI et al., 2024) suggest an alternative: *in-context learning* (ICL; Dong et al., 2024), where models learn new tasks directly from examples provided as input, without parameter up-

[1] Code and data are available at `https://github.com/bflashcp3f/deer`.

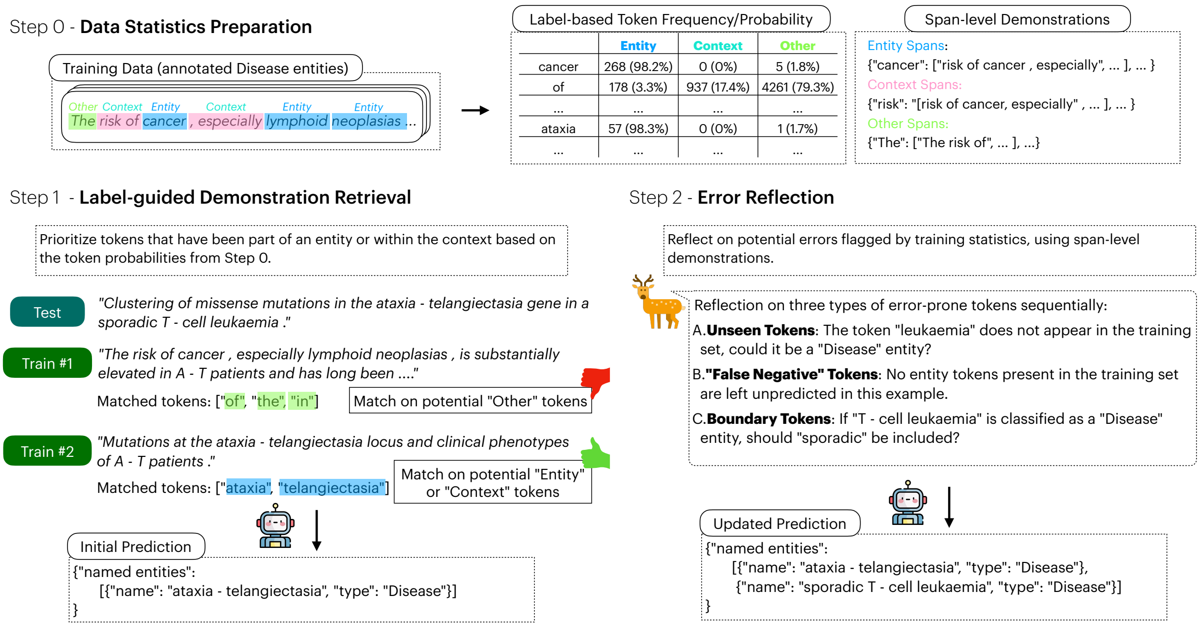
Step 0 - Data Statistics Preparation
Training Data (annotated Disease entities)
Other Context Entity Context Entity Entity
The risk of cancer , especially lymphoid neoplasias ...
Label-based Token Frequency/Probability
Entity Context Other
cancer 268 (98.2%) 0 (0%) 5 (1.8%)
of 178 (3.3%) 937 (17.4%) 4261 (79.3%)
... ...
ataxia 57 (98.3%) 0 (0%) 1 (1.7%)
... ...
Span-level Demonstrations
Entity Spans:
{"cancer": ["risk of cancer , especially", ...], ... }
Context Spans:
{"risk": "[risk of cancer, especially" , ...], ... }
Other Spans:
{"The": ["The risk of", ...], ...}
Step 1 - Label-guided Demonstration Retrieval
Prioritize tokens that have been part of an entity or within the context based on the token probabilities from Step 0.
Test
"Clustering of missense mutations in the ataxia - telangiectasia gene in a sporadic T - cell leukaemia ."
Train #1
"The risk of cancer , especially lymphoid neoplasias , is substantially elevated in A - T patients and has long been ...."
Matched tokens: ["of", "the", "in"]
Match on potential "Other" tokens
Train #2
"Mutations at the ataxia - telangiectasia locus and clinical phenotypes of A - T patients ."
Matched tokens: ["ataxia", "telangiectasia"]
Match on potential "Entity" or "Context" tokens
Initial Prediction
{"named entities":
[{"name": "ataxia - telangiectasia", "type": "Disease"}]
}
Step 2 - Error Reflection
Reflect on potential errors flagged by training statistics, using span-level demonstrations.
Reflection on three types of error-prone tokens sequentially:
A.Unseen Tokens: The token "leukaemia" does not appear in the training set, could it be a "Disease" entity?
B."False Negative" Tokens: No entity tokens present in the training set are left unpredicted in this example.
C.Boundary Tokens: If "T - cell leukaemia" is classified as a "Disease" entity, should "sporadic" be included?
Updated Prediction
{"named entities":
[{"name": "ataxia - telangiectasia", "type": "Disease"},
{"name": "sporadic T - cell leukaemia", "type": "Disease"}]
}

Figure 2: Overview of DEER. In the preparation stage (Step 0), the method compiles training input and labels to compute token frequencies and probabilities in three scenarios: 1) entity token, 2) context token, and 3) other token, along with their associated spans. In the inference stage (Step 1 and 2), Step 1 retrieves sentence-level demonstrations by emphasizing potential entity- and context-related tokens based on probabilities from Step 0. Step 2 refines predictions from Step 1 by addressing error-prone tokens based on label statistics, focusing on three token types: unseen tokens, "false negative" tokens, and boundary tokens. For each token type, the refinement process retrieves span-level demonstrations and prompts LLMs to adjust predictions. See §2 for further details.

dates. As shown in Figure 1, a common ICL strategy for NER (Wang et al., 2023a; Lu et al., 2024) involves retrieving a fixed number of demonstrations using embeddings of input data, and then prompting the model to extract entities. While such methods show particular promise in low-resource scenarios (Lee et al., 2022), it does not scale as efficiently as supervised methods (Sainz et al., 2024). For example, Monajatipoor et al. (2024) report a 79.3 F_1 score on the NCBI dataset using GPT-4 with 16 examples per query. In contrast, a fine-tuned BERT-base model achieves an 84.0 F_1 score when given access to the same training data.

In this paper, we argue that this performance gap primarily stems from inefficiencies in demonstration retrieval rather than from inherent limitations of LLMs. The prevailing sentence embedding–based retrievers are ill-suited for NER for two key reasons: (1) NER is a token-level task requiring fine-grained sensitivity to entity boundaries, whereas sentence embeddings operate at a coarse, sentence-level granularity; and (2) these embeddings are task-agnostic, disregarding label information from the training data, even though an ideal retriever should recognize entity-focused patterns and retrieve demonstrations containing similar entities to those in the query. As shown in Table 6, our sanity-check analysis confirms that sentence embedding–based retrievers frequently fail to retrieve demonstrations containing overlapping entities with the query, highlighting their inefficiency.

Motivated by these findings, we propose DEER (**D**ata Stat.-grounded nam**E**d **E**ntity **R**ecognition), a method designed to help LLMs make more informed predictions during ICL in a training-free manner. The core idea of DEER is to leverage task-specific information embedded in label statistics. In the context of NER, this corresponds to identifying which tokens are more likely to appear within entity spans, near entity spans, or outside them. Based on such statistics from the entire training set, DEER incorporates them in two complementary stages. First, the retriever uses token-level statistics to guide demonstration selection, prioritizing examples containing tokens that are likely to correspond to entities or their contexts, resulting in more relevant demonstrations. Second, during an error reflection stage, DEER revisits potentially misclassified tokens (flagged by these statistics) with targeted span-level demonstrations, enabling the model to refine its predictions and achieve improved overall performance.

We evaluate DEER on five NER datasets spanning four domains: news, biomedicine, social media, and mixed web text, using four LLMs: `Qwen2.5-7B`, `Llama3.3-70B`, `GPT-4o-mini`, and `GPT-4o`. The results show that DEER consistently outperforms ICL baselines across datasets and models. With `GPT-4o`, DEER achieves performance competitive with supervised models on four out of five datasets, while `Llama3.3-70B` also demonstrates strong results. Further analysis confirms that DEER enhances example retrieval, improves recognition of both seen and unseen entities, and demonstrates robustness in low-resource settings.

2 DEER 🦌

We introduce **DEER**, an ICL approach that enables LLMs to make more informed entity predictions through the use of label statistics. As illustrated in Figure 2, DEER consists of two main components: (1) label-guided retrieval and (2) error reflection. The following sections describe the problem setup and the construction of the label statistics in DEER, followed by explanations of each component.

2.1 Problem Setup

Named entity recognition aims to identify all text spans corresponding to pre-defined entity types in a sentence. Following prior work on LLM-based entity recognition (Dunn et al., 2022; Lu et al., 2022), we formulate the task as a span identification problem and format the output as a structured JSON object (Figure 1), using the key *"named_entities"* for a list of JSON objects, each containing the name and type of an entity (e.g., *{"name": "Barack Obama", "type": "PERSON"}*). In Appendix B, we compare it with two other prompting formats from recent work and find that our chosen format yields the best performance.

2.2 Data Statistics Preparation

As motivated in §1, an effective demonstration retriever should capture entity-focused patterns and retrieve demonstrations that exhibit similar patterns to those in the query. To achieve this, we leverage label-grounded token statistics from the training data. Specifically, we categorize all training tokens into three classes based on their relation to entities and compute the frequency and probability of each token within these classes (Step 0 in Figure 2).

Entity tokens are those that belong to an identified entity. Context tokens are the tokens surrounding an entity, typically two tokens on each side, which provide identification cues and help define entity boundaries (Huang et al., 2015). For example, consider the sentence in Figure 2, "`The risk of cancer, ...`", where "cancer" is an entity token, and "risk" and "of" are context tokens. Context tokens like "risk of" provide contextual clues for identifying Disease entities, even in cases where entities are unseen in the training data. Finally, **other tokens** include all remaining tokens that fall outside the entity and its immediate context window (e.g., "The" in the example).

Moreover, we define token-focused spans for these categories: entity spans include the entity along with its surrounding context, context spans are similar but correspond to context tokens, and other spans consist of two adjacent tokens on each side of an other token. For instance, the span "`risk of cancer, especially`" serves as an entity span for "cancer" but as a context span for "risk."[2] These label-dependent spans offer a more precise and structured approach to handling tokens in different scenarios. In §4.1, we analyze the impact of incorporating these three token types and demonstrate the importance of context tokens, particularly for improving performance on unseen entities.

2.3 Label-guided Retrieval

DEER begins with label-guided, token-based retrieval for ICL (Step 1 in Figure 2). The ICL prompt consists of the task description, specifying the targeted entity types and output format, followed by N examples selected from the training set and the query sentence. Unlike previous work that uses task-agnostic sentence embeddings, we aim to address the token-centric nature of NER. Specifically, to compute the similarity between a training sentence s_i and a test sentence s_t, we define a similarity score $\mathbf{S}_{s_i,s_t}$ that combines exact token matching and embedding-based semantic similarity:

$$\mathbf{S}_{s_i,s_t} = \lambda_1 \cdot S_{\text{token}}(s_i, s_t) + \lambda_2 \cdot S_{\text{embed}}(s_i, s_t), \quad (1)$$

where λ_1 and λ_2 are hyper-parameters controlling the relative contribution of two components. The token match score S_{token} is computed as:

[2] In our implementation, entity spans exclude neighboring entities within the context window to encourage greater attention on context tokens for more generalizable entity recognition. For example, given the sentence "Lionel Messi and Cristiano Ronaldo are exceptional football players" and a context window of two, the entity spans for the two PERSON entities are "`Lionel Messi and`" and "`and Cristiano Ronaldo are exceptional`".

$$S_{\text{token}}(s_i, s_t) = \sum_{t \in s_t} \mathbb{I}(t \in s_i) \cdot W(t), \tag{2}$$

where $\mathbb{I}(t \in s_i)$ is an indicator function that equals 1 if test token t appears in the training sentence s_i, and 0 otherwise. The function $W(t)$ assigns importance to each token:

$$W(t) = \begin{cases} w_e P(t_e) + w_c P(t_c) + w_o P(t_o), & \text{if } t \text{ is seen,} \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

where $P(t_e)$, $P(t_c)$, and $P(t_o)$ are the probabilities of token t being an entity, context, or other token, respectively, in the training corpus. The hyperparameters w_e, w_c, and w_o control the relative importance of these token types, allowing the model to prioritize entity and context tokens, which are more informative for NER.[3] For semantic similarity, we compute a sentence embedding by aggregating token embeddings:

$$\mathbf{v}_s = \sum_{t \in s} W(t) \cdot \mathbf{v}_t, \tag{4}$$

where $\mathbf{v}_t$ represents the embedding of token t. The final score is computed via cosine similarity:

$$S_{\text{embed}}(s_i, s_t) = \cos(\mathbf{v}_{s_i}, \mathbf{v}_{s_t}). \tag{5}$$

By integrating both token matching and embedding-based similarity, our method ensures precise retrieval of seen tokens while also accounting for unseen tokens, leading to optimized performance across different dataset characteristics.

2.4 Error Reflection (ER)

Even with improved retrieval, ICL can only incorporate a small subset of training examples, leaving blind spots for LLM predictions. For example, the model may miss entities present in the training set but absent from retrieved examples or make boundary errors due to insufficient guidance. To address this, we introduce an error reflection step that uses prepared statistics to identify error-prone tokens and provides targeted demonstrations to help the model revise its predictions. We focus on single-entity, boundary-related errors, which have been identified as a major source of NER mistakes (Ding et al., 2024; Lu et al., 2024). A hierarchical breakdown of NER errors is shown in Figure 4.

As shown in Step 2 of Figure 2, our reflection step targets three token categories associated with potential errors: unseen tokens, “false negative” tokens, and boundary tokens of predicted entities. The first two categories help recover missed entities, while the third addresses both false positives and boundary-related errors. The reflection process proceeds sequentially. We first examine unseen and false negative tokens in the query sentence to identify additional candidate entities. Then, we refine or discard predictions by inspecting boundary tokens. To enable this process, we use a chain-of-thought reasoning approach (Wei et al., 2023), prompting the model to reason about the relationship between the query and retrieved demonstrations before making final predictions. Details on how we identify these tokens and construct reflection prompts for each category are provided below. Exact prompts are listed in Appendix A.1.

Unseen Tokens Unseen tokens, defined as tokens not present in the training data, often lead to false negatives since they lack direct matches for retrieval. To address this, we leverage their surrounding context. Specifically, we select unseen tokens that were not predicted as part of an entity but are surrounded by tokens frequently labeled as entities or context tokens in the training set. This targeted strategy avoids reflecting on every unseen token, reducing inference cost while improving prediction accuracy (see Table 12 in the appendix for reflection frequency statistics). For each selected unseen token, we retrieve M span-level demonstrations from the training set by matching its context tokens. A chain-of-thought prompt then guides the model to reason through these demonstrations and determine whether a new entity should be extracted. For sentences with multiple unseen tokens, this process is applied in the order the tokens appear.

“False Negative" Tokens These are tokens commonly annotated as entities in the training set but missed during initial predictions. A token is flagged as a “false negative" if its entity likelihood in the training set exceeds a predefined threshold. For each such token, we retrieve M span-level examples containing it and prompt the model to predict whether an entity should be extracted.

Boundary Tokens These tokens lie at the edges of predicted entities and are critical for refining entity spans or eliminating false positives. For each prediction, we examine $2K$ tokens around

[3] For all hyperparameters, we perform a grid search over three possible values: $[1.0, 0.5, 0.01]$, and find that setting $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $w_e = 1.0$, $w_c = 1.0$, and $w_o = 0.01$ performs well in most cases. See ablations in Appendix B.

the boundary: K edge tokens of the entity and K from the surrounding context. We focus on tokens likely to be misclassified based on training statistics, for example, tokens predicted as part of an entity but appeared more as context tokens in the training set. For each reflected token, we retrieve span-level examples from all three categories: entity, context, and other. This variety allows the model to reason through different scenarios and decide whether to include each token, ultimately refining the entity's boundaries.

3 Experimental Setup

3.1 Dataset and Evaluation Metric

To demonstrate the effectiveness of our method, we evaluate it on five NER datasets spanning four domains, which have been used in recent ICL-based NER studies (Wang et al., 2023a; Monajatipoor et al., 2024). These datasets include CoNLL03 (Sang and De Meulder, 2003) for news, NCBI-disease (Doğan et al., 2014) and bc2gm (Smith et al., 2008) for biomedicine, TweetNER7 (Ushio et al., 2022) for social media, and OntoNotes (Weischedel et al., 2013) for web text from various sources. Dataset statistics are provided in Table 1. We refer readers to the corresponding papers for details on each dataset. Following prior work (Berger et al., 2024; Bai et al., 2024), we control computational costs by limiting the test set size to 1,000 examples through random sampling of their original test sets. To ensure the sampled subset provides a reliable estimate of model performance, we calculate 95% confidence intervals using 1,000 bootstrap resamples, and the margins of error fall within a 0.01-0.03 range, verifying the validity of our performance estimates. Further details on this process can be found in Appendix A.2.

All experiments are evaluated using the standard mention-level matching metric for NER (Sang and De Meulder, 2003), where a predicted entity is considered correct only if it matches both the span boundary and the entity type.

3.2 Baselines

We compare DEER against both ICL and fine-tuning baselines to demonstrate its improvements over standard ICL methods and its ability to approach fine-tuning performance. All methods use the full training set for either fine-tuning or in-context demonstration selection.

Dataset	Domain	# Ent.	# Sentences (Tr / Dv / Ts)
CoNLL03 (2003)	News	4	14.0k / 3.3k / 3.5k
OntoNotes (2013)	Web	18	60.0k / 8.5k / 8.3k
NCBI (2014)	Biomed	1	5.4k / 0.9k / 0.9k
BC2GM (2008)	Biomed	1	12.5k / 2.5k / 5.0k
TweetNER7 (2022)	Social	7	7.1k / 0.8k / 0.6k

Table 1: Statistics of the five experimented datasets. Each dataset contains a minimum of 5,000 annotated training examples, ensuring competitive performance of fine-tuned models.

ICL Baselines We cover three ICL baselines: KATE (Liu et al., 2022), BM25 (Robertson et al., 2009), and BSR (Gupta et al., 2023). **KATE** serves as our primary ICL baseline, as it remains the dominant framework for ICL-based NER (Wang et al., 2023a; Berger et al., 2024; Monajatipoor et al., 2024). This method selects in-context examples through nearest neighbor search using sentence-level embeddings. **BM25** treats both the test input and demonstrations as bags of words and measures their relevance via a TF-IDF–weighted recall of overlapping words. Recent IE work (Sun et al., 2024) has shown its effectiveness for demonstration retrieval. **BSR** employs BERTScore-Recall (Zhang et al., 2020) as a similarity metric, which measures token-level recall using token embeddings. While mainly used in semantic parsing, we adapt it here for NER. To ensure the competitiveness of the baselines, we evaluate seven embedding models commonly used in recent studies (see Appendix B). OpenAI's `text-embedding-3-small` consistently outperforms open-source alternatives such as `all-mpnet-base-v2`, and is therefore used in all our experiments. We do not include **GPT-NER** (Wang et al., 2023a), an entity-based retriever, due to its impracticality for the ICL setting. Nonetheless, we show in Table 14 that our method outperforms GPT-NER. Additional discussion on GPT-NER can be found in §5.

Fine-tuning Baselines We compare with five supervised models: BERT-base, UniversalNER-7B (Zhou et al., 2024), InstructUIE-11B (Wang et al., 2023b), and GoLLIE-13B/34B (Sainz et al., 2024). **BERT-base** (Devlin et al., 2019) is fine-tuned for token-level classification; we run three random seeds and report the averaged performance. **UniversalNER-7B** applies targeted distillation for NER by prompting `gpt-3.5-turbo` to annotate unlabeled text, training a student model on the refined data, and subsequently fine-tuning it on super-

LLM	Method	NCBI	bc2gm	CoNLL03	OntoNotes	TwNER7	average
Qwen2.5-7B	BM25	66.8	57.0	83.0	62.6	55.0	64.9
	BSR (2023)	68.0	59.9	86.0	66.8	56.1	67.4
	KATE (2022)	66.9	59.3	83.7	63.8	57.1	66.2
	DEER$_{\text{w/o ER}}$	71.7	62.5	87.2	68.4	57.3	69.4
	DEER	**73.2**	**63.6**	**87.8**	**71.8**	**57.5**	**70.8**
Llama3.3-70B	BM25	79.7	65.5	88.9	76.0	61.7	74.4
	BSR (2023)	79.3	67.7	90.2	78.9	62.8	75.8
	KATE (2022)	79.4	66.5	88.3	76.1	62.0	74.5
	DEER$_{\text{w/o ER}}$	81.9	68.6	89.6	78.9	63.3	76.5
	DEER	**83.9**	**69.6**	**90.2**	**80.2**	**63.5**	**77.5**
GPT-4o-mini	BM25	74.1	62.9	87.8	74.4	59.3	71.7
	BSR (2023)	74.9	62.9	89.6	77.4	61.1	73.2
	KATE (2022)	76.0	64.1	89.2	74.9	59.6	72.8
	DEER$_{\text{w/o ER}}$	77.3	65.4	89.9	77.0	60.4	74.1
	DEER	**79.3**	**67.1**	**90.9**	**79.2**	**61.6**	**75.8**
GPT-4o	BM25	81.0	71.9	92.0	81.2	62.8	77.8
	BSR (2023)	81.2	73.5	93.3	82.9	63.9	78.9
	KATE (2022)	80.0	73.6	92.9	80.7	62.2	77.9
	DEER$_{\text{w/o ER}}$	82.7	74.3	93.5	83.5	63.6	79.5
	DEER	**84.8**	**75.4**	**93.6**	**85.2**	**64.3**	**80.7**

Table 2: Comparison of DEER with three ICL baselines across five datasets using four recent LLMs and 8 demonstrations per query. DEER$_{\text{w/o ER}}$ excludes the error reflection step and already outperforms all baselines, validating the effectiveness of our label-guided retrieval. Adding error reflection yields further improvements. Among the LLMs, GPT-4o performs best overall, while Llama3.3-70B surpasses GPT-4o-mini and approaches GPT-4o, demonstrating the promise of open-source models.

			Unseen Entity			
	Seen Entity		Seen Token		Unseen Token	
	KATE	DEER	KATE	DEER	KATE	DEER
NCBI	88.7	**91.4**	61.1	**66.5**	70.3	**81.0**
bc2gm	85.8	**88.4**	**60.5**	60.1	70.8	**75.0**
CoNLL03	95.5	**95.7**	84.9	**87.7**	91.2	**92.1**
OntoNotes	86.0	**89.6**	65.5	**72.4**	76.1	**79.7**
TweetNER7	71.1	**74.9**	45.9	**46.9**	58.3	**58.3**

Table 3: Breakdown of test set performance for KATE and DEER (with GPT-4o) based on entity presence in the training data. Unseen entities are further categorized by whether they contain novel tokens. DEER outperforms KATE on both seen and unseen entities, highlighting its effectiveness to novel entities.

vised NER datasets.[4] **InstructUIE-11B** adopts unified instruction tuning, training FlanT5-11B (Chung et al., 2022) on 32 diverse IE datasets reformulated into a text-to-text format with expert-written instructions, yielding strong performance in both supervised and low-resource settings. Finally, **GoLLIE-13B** and **GoLLIE-34B** extend CodeLlama (Rozière et al., 2024) with human-authored guidelines, leading to substantial improvements in zero-shot IE while also maintaining competitive supervised results.

3.3 Implementation Details

We conduct experiments with four recent LLMs, covering both open-source and proprietary models: Qwen2.5-7B (Qwen et al., 2025), Llama3.3-70B (Dubey et al., 2024), GPT-4o-mini, and GPT-4o (OpenAI et al., 2024).[5] We use the same prompt across four LLMs and set the decoding temperature to 0 to facilitate result reproducibility. For main ICL experiments, we retrieve eight demonstrations per query, sorted in ascending order of similarity to the test example. Appendix A contains detailed descriptions of the prompts and all hyperparameters.

4 Results

Comparison with ICL Baselines Table 2 presents the results of three ICL baselines and two variants of DEER, including the one without error reflection (DEER$_{\text{w/o ER}}$), evaluated across five datasets and four LLMs. DEER$_{\text{w/o ER}}$ consistently

[4] As noted in its GitHub repository, the authors have not released the code or prompts necessary to reproduce most supervised results due to licensing restrictions. We therefore rely on the reported results in the original paper.

[5] gpt-4o-mini-2024-07-18 and gpt-4o-2024-08-06 are used through OpenAI API. In our preliminary experiments, we also evaluated Llama3.1-8B. However, it failed to generate outputs in the required format during ICL (see Appendix B.6 for more details). Thus, we did not pursue it further.

	8 demo.		32 demo.		**Fine-tuned**				
Dataset	DEER$_{\text{w/o ER}}$	DEER	DEER$_{\text{w/o ER}}$	DEER	BERT	UniNER$_{7B}$	InsUIE$_{11B}$	GoLLIE$_{13B}$	GoLLIE$_{34B}$
NCBI	82.7 ($2.1)	84.8 ($3.0)	84.1 ($6.5)	85.7 ($7.3)	84.0	87.0	86.2	86.5	85.8
bc2gm	74.3 ($2.9)	75.4 ($3.7)	75.0 ($9.4)	75.6 ($9.9)	79.0	82.4	80.7	-	-
CoNLL03	93.5 ($1.9)	93.6 ($2.2)	93.9 ($5.8)	94.0 ($6.0)	90.4	93.3	91.5	93.0	93.1
OntoNotes	83.5 ($2.5)	85.2 ($3.3)	85.8 ($7.5)	86.6 ($8.2)	87.2	89.9	88.7	84.0	84.6
TweetNER7	63.6 ($1.9)	64.3 ($3.4)	64.9 ($6.1)	65.3 ($7.2)	60.9	65.7	66.0	-	-

Table 4: ICL results of DEER (using GPT-4o) with 8 and 32 demonstrations, compared to five fine-tuning baselines. Values in parentheses indicate associated API inference costs on the test set. Increasing the number of demonstrations improves performance, and error reflection continues to provide additional gains at larger scales. Notably, DEER with 8-shot error reflection performs competitively with the 32-shot setting without reflection, but at a significantly lower cost. DEER also matches or exceeds fine-tuning baselines on four out of five datasets.

outperforms KATE in all settings, demonstrating the effectiveness of our label-guided, token-level demonstration retrieval. It also performs better than BM25 and BSR. Interestingly, BSR consistently outperforms KATE, suggesting that BERTScore-Recall, which emphasizes token-level coverage, may be more suitable than cosine similarity on sentence representations for token-centric tasks like NER. A promising future direction would be to integrate our label-guided retrieval with BERTScore-Recall. Adding error reflection leads to further gains across all settings, confirming its effectiveness. A breakdown of performance across the three reflection steps is provided in Table 8 in the appendix. Appendix C (Table 20) presents an error analysis, highlighting DEER's advantages in reducing under-predicted entities and boundary errors.

Among the LLMs, GPT-4o achieves the best overall results while our method delivers larger relative improvements on smaller models, such as Qwen2.5-7B. For example, on OntoNotes, DEER achieves an 8.0 absolute F_1 gain over KATE. Llama3.3-70B shows strong performance, outperforming GPT-4o-mini and approaching GPT-4o, highlighting the potential of open-source LLMs.

Table 3 further analyzes performance based on entity presence in the training data. The test set entities are categorized as either seen or unseen during training. Unseen entities are further divided into those containing novel tokens and those formed by permutations of seen tokens. The results show that DEER consistently outperforms KATE on both seen and unseen entities, demonstrating its effectiveness in generalizing to novel entities.

Scaling Demonstrations and Comparison with Fine-tuning Baselines To evaluate the scalability of DEER and its performance relative to fine-tuning, Table 4 reports results using GPT-4o with 8 and 32 demonstrations per query, along with five fine-tuning baselines. We include both performance and API inference costs, with and without error reflection.[6] Increasing the number of demonstrations to 32 consistently improves performance across all datasets. Error reflection continues to provide added benefit even at this larger scale. Notably, DEER with error reflection and 8 demonstrations performs competitively with the 32-demo version without reflection, but at significantly lower cost. This highlights a potential cost-performance tradeoff between scaling demonstrations and applying error reflection, which we leave for future work.

Compared to five fine-tuning baselines, DEER achieves comparable or superior performance on four out of five datasets, e.g., surpassing GoLLIE-34B on CoNLL03 and OntoNotes. These results suggest that the reliance on task-specific NER models may diminish, as general-purpose LLMs become increasingly cost-effective and reusable across tasks, while specialized models incur additional maintenance and deployment overheads.

4.1 Ablation Studies & Analyses

Label-guided Retriever To better understand the effectiveness of our retriever, we compare three retrieval methods across all five datasets using GPT-4o: sentence embeddings from KATE, our proposed token-based retriever, and an unweighted variant that excludes label-guided token statistics. All embeddings are generated using OpenAI's text-embedding-3-small model. As shown in Table 5, our label-guided method consistently outperforms the other two. The unweighted variant performs similarly to or worse than sentence embeddings, underscoring the importance of incor-

[6] Inference costs are calculated using OpenAI's official pricing: $2.50 per million input tokens and $10.00 per million output tokens.

porating label-grounded, token-level weighting in retrieval. Table 6 further provides a sanity check on demonstration retrieval quality: for entities that appear in both the training and test sets, our retriever includes a substantially higher proportion of these entities within the retrieved demonstrations compared to task-agnostic sentence embeddings, validating our argument about the inefficiency of generic retrievers. Further analysis of the two components in our retriever is provided in Appendix B.

Dataset	Sentence	Token (uwg.)	Token (wg, ours)
NCBI	80.0	80.2	**82.7**
bc2gm	73.6	71.6	**74.3**
CoNLL03	93.1	92.6	**93.5**
OntoNotes	80.7	81.8	**83.5**
TweetNER7	62.2	62.7	**63.6**

Table 5: Comparison of three demonstration retrievers: sentence embeddings (KATE), our label-guided token-based retriever, and an unweighted variant without label statistics. Embeddings are generated using OpenAI's `text-embedding-3-small`. Our method consistently outperforms the others, highlighting the value of label-grounded, token-level weighting.

Dataset	Sentence	Token (wg, ours)
NCBI	85.0	93.3
bc2gm	72.5	83.1
CoNLL03	70.1	74.6
OntoNotes	62.7	80.6
TweetNER7	76.4	77.6

Table 6: Sanity check for demonstration retrievers. For entities appearing in both the training and test sets, we compare different retrievers based on the proportion of these entities contained within the retrieved eight demonstrations. Our retriever yields substantially more relevant entities than task-agnostic sentence embedding baselines, supporting our argument regarding the inefficiency of generic retrievers.

Token Types & Context Length A key contribution of DEER is the introduction of three token types, particularly context tokens, defined as tokens surrounding annotated entities within a specified context window. We conduct an ablation study on different context lengths, ranging from 0 to 3, where 0 indicates the removal of context tokens, leaving tokens classified only as entity or other tokens. The ICL results of DEER on three datasets, presented in Figure 3, show that incorporating context tokens significantly boosts performance, although the improvement plateaus once the context length reaches 2. Moreover, Table 9 in the appendix provides a breakdown of performance on seen and unseen entities as in Table 3, confirming that context tokens are effective in improving performance on unseen entities, supporting our hypothesis.

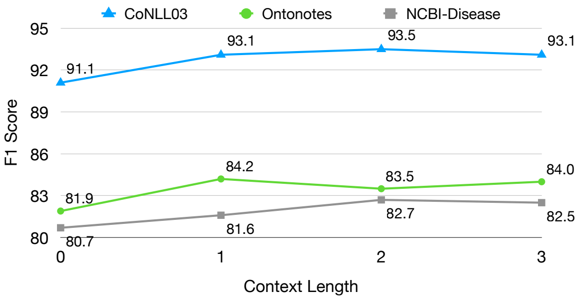

Figure 3: Comparison of different context lengths in DEER across three datasets, where 0 indicates that tokens are classified as either entity or other tokens. The substantial performance gain from context length 1 over 0 highlights the effectiveness of incorporating context tokens in our approach.

Low-Resource Results Thus far, our evaluation has focused on high-resource settings, where training sets contain at least 5,000 examples. To assess DEER's effectiveness in low-resource scenarios, we downsample NCBI to three smaller scales: 16, 64, and 256 examples, and compare its performance against KATE. The results in Table 7 show that DEER consistently outperforms KATE, demonstrating its robustness and effectiveness in low-resource conditions.

Method	16-shot	64-shot	256-shot
KATE	70.3	73.2	73.6
DEER	**75.7**	**76.0**	**78.1**

Table 7: Low-resource experiments on NCBI. We compare DEER with KATE (using `GPT-4o`) across three scales: 16, 64, and 256 examples. DEER surpasses KATE in all settings, highlighting its effectiveness in low-resource settings.

5 Related Work

LLM-Based Named Entity Recognition The rise of LLMs has opened new avenues for NER by leveraging their capabilities in instruction-following (Ouyang et al., 2022) and in-context learning (Brown et al., 2020). Recent work on LLM-based NER follows two main directions. One focuses on training NER-specialized models by fine-tuning open-source LLMs (Wang et al., 2023b; Zhou et al., 2024; Sainz et al., 2024). The other, which we follow, explores adapting general-purpose LLMs for NER via prompting (Xie et al., 2023; Ashok and Lipton, 2023; Han et al., 2024). In this direction, the dominant framework, KATE (Liu et al., 2022), selects demonstrations using sentence embeddings. Prior research mainly varies prompting templates (Li et al., 2023) or embedding models for retrieval (Monajatipoor et al., 2024), without addressing the core limitations of KATE in leveraging labeled data for token-level tasks. We address this gap by introducing a label-guided retriever and an error reflection strategy. The most relevant prior work is Wang et al. (2023a), which proposes an entity-based retriever that requires training a BERT-based NER model to guide ICL retrieval. However, the resulting ICL model underperforms the BERT-based retriever itself, limiting its practical value. In contrast, our method shows that general-purpose LLMs can achieve competitive performance with supervised models in a training-free manner.

Reflection Using LLMs An emerging capability of LLMs is their ability to reason (Wei et al., 2023) and self-reflect (Asai et al., 2023) on their predictions, enabling them to correct errors. Recent research has explored this capability across a wide range of tasks, like code generation (Madaan et al., 2023) and planning (Shinn et al., 2023). Similar approaches have also been applied to NER (Wang et al., 2023a; Chen et al., 2024). These works primarily rely on LLMs' internal knowledge for decision-making. In contrast, our approach grounds model predictions to the label-grounded training statistics. This grounding enables the model to capture fine-grained dataset-specific artifacts, akin to supervised methods, and helps close the performance gap between prompting-based and supervised approaches.

6 Conclusion

We present **DEER**, a training-free ICL method designed to help LLMs make more informed entity predictions. DEER leverages label statistics to identify tokens that are most informative for entity recognition, thereby improving demonstration retrieval and enabling effective error reflection. Evaluated on five NER datasets across four LLMs, DEER consistently outperforms existing ICL approaches and achieves performance comparable to supervised fine-tuning. Further analyses highlight the limitations of commonly used sentence embedding-based retrievers and demonstrate DEER's effectiveness on both seen and unseen entities, as well as its robustness in low-resource settings.

Limitations

This paper focuses on flat NER while acknowledging that other NER challenges, such as discontinuous NER (Dai et al., 2020) and nested NER (Finkel and Manning, 2009), pose unique difficulties, particularly for autoregressive LLMs. Exploring these more complex NER tasks is left for future work. Our method primarily targets demonstration selection within the in-context learning framework. Future research could extend this by jointly optimizing both demonstration selection and task instructions (Pang et al., 2023), as incorporating richer task-specific prompts may improve generalization in more difficult scenarios. Finally, the error reflection mechanism used in this work is guided by domain knowledge and limited to three predefined error types. Recent studies have shown that LLMs are capable of detecting their own errors (Wang et al., 2024; Kamoi et al., 2024), suggesting the potential for automating error reflection using more fine-grained or adaptive error categories.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Dhananjay Ashok and Zachary C. Lipton. 2023. Promptner: Prompting for named entity recognition. *Preprint*, arXiv:2305.15444.

Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. 2024. Schema-driven information extraction from heterogeneous tables. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10252–10273, Miami, Florida, USA. Association for Computational Linguistics.

Fan Bai, Alan Ritter, Peter Madrid, Dayne Freitag, and John Niekrasz. 2022. SynKB: Semantic search for synthetic procedures. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 311–318, Abu Dhabi, UAE. Association for Computational Linguistics.

Fan Bai, Alan Ritter, and Wei Xu. 2021. Pre-train or annotate? domain adaptation with a constrained budget.

In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5002–5015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Uri Berger, Tal Baumel, and Gabriel Stanovsky. 2024. In-context learning on a budget: A case study in named entity recognition. *Preprint*, arXiv:2406.13274.

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.

Wei Chen, Lili Zhao, Zhi Zheng, Tong Xu, Yang Wang, and Enhong Chen. 2024. Double-checker: Large language model as a checker for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3172–3181, Miami, Florida, USA. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5860–5870, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Yan Bowen, and Min Zhang. 2024. Rethinking negative instances for generative named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3461–3475, Bangkok, Thailand. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S. Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. 2022. Structured information extraction from complex scientific text with fine-tuned large language models. *Preprint*, arXiv:2212.05238.

Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL HLT*

2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 80–88, Los Angeles. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 141–150.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13924–13950, Singapore. Association for Computational Linguistics.

Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. 2024. An empirical study on information extraction using large language models. *Preprint*, arXiv:2305.14450.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *Preprint*, arXiv:1508.01991.

Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin Lu, Nan Zhang, Yusen Zhang, Ranran Haoran Zhang, Sujeeth Reddy Vummanthala, Salika Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. 2024. Evaluating llms at detecting errors in llm responses. *Preprint*, arXiv:2404.03602.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022. Good examples make a faster learner: Simple demonstration-based learning for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2700, Dublin, Ireland. Association for Computational Linguistics.

Ulf Leser and Jörg Hakenberg. 2005. What makes a gene name? named entity recognition in the biomedical literature. *Briefings in bioinformatics*, 6(4):357–369.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023. CodeIE: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Toronto, Canada. Association for Computational Linguistics.

Bill Yuchen Lin and Wei Lu. 2018. Neural adaptation layers for cross-domain named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2012–2022, Brussels, Belgium. Association for Computational Linguistics.

Yi-Feng Lin, Tzong-Han Tsai, Wen-Chi Chou, Kuen-Pin Wu, Ting-Yi Sung, and Wen-Lian Hsu. 2004. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th International Conference on Data Mining in Bioinformatics*, pages 56–61. Citeseer.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Shuheng Liu and Alan Ritter. 2023. Do CoNLL-2003 named entity taggers still work well in 2023? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8254–8271, Toronto, Canada. Association for Computational Linguistics.

Qiuhao Lu, Rui Li, Andrew Wen, Jinlian Wang, Liwei Wang, and Hongfang Liu. 2024. Large language models struggle in token-level clinical named entity recognition. *Preprint*, arXiv:2407.00731.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Masoud Monajatipoor, Jiaxin Yang, Joel Stremmel, Melika Emami, Fazlolah Mohaghegh, Mozhdeh Rouhsedaghat, and Kai-Wei Chang. 2024. Llms in biomedicine: A study on clinical named entity recognition. *Preprint*, arXiv:2404.07376.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389, Singapore. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *Preprint*, arXiv:2101.05779.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code. *Preprint*, arXiv:2308.12950.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. *Preprint*, arXiv:2310.03668.

Erik F. Sang, Tjong Kim 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.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9:1–19.

Zhaoyue Sun, Gabriele Pergola, Byron Wallace, and Yulan He. 2024. Leveraging ChatGPT in pharmacovigilance event extraction: An empirical study. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 344–357, St. Julian's, Malta. Association for Computational Linguistics.

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.

Asahi Ushio, Francesco Barbieri, Vitor Sousa, Leonardo Neves, and Jose Camacho-Collados. 2022. Named entity recognition in Twitter: A dataset and analysis on short-term temporal shifts. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 309–319, Online only. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models. *Preprint*, arXiv:2304.10428.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. 2024. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *The Twelfth International Conference on Learning Representations*.

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.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, Ann Houston, et al. 2013. Ontonotes release 5.0. *LDC2013T19, Philadelphia, Penn.: Linguistic Data Consortium*, 17.

Casey Whitelaw, Alex Kehlenbeck, Nemanja Petrovic, and Lyle Ungar. 2008. Web-scale named entity recognition. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 123–132.

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

Tianyi Zhang, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

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

A Implemetation Details

A.1 DEER Prompts

We use the CoNLL03 dataset (Sang and De Meulder, 2003), which targets the identification of "Person," "Organization," "Location," and "Miscellaneous" entities, as an example to illustrate the prompts used in our experiments.

In-Context Learning - Input

```
Here is the JSON template for named entity
    recognition:
{"named entities": [{"name": "ent_name_1", "type
    ": "ent_type_1"}, ..., {"name": "ent_name_n
    ", "type": "ent_type_n"}]}

Please identify the four types of named entities:
     "PER", "LOC", "ORG", and "MISC" ("
    Miscellaneous"), following the JSON template
     listed above, and output the JSON object.
    If no named entities identified, output {"
    named entities": []}.

Input: The girl , who was accompanied to
    Philadelphia by her parents , will need more
     surgery later to correct the condition on
    her chest , back and legs , the hospital
    said .
Output: {"named entities": [{"name": "
    Philadelphia", "type": "LOC"}]}

Input: " I know what I 'm here for , " said
    Medvedev , who lost in the second round of
    the Open the last two years after reaching
    the quarters in 1993 , the same year he
    tried his hand as a restaurant critic .
Output: {"named entities": [{"name": "Medvedev",
     "type": "PER"}, {"name": "Open", "type": "
    MISC"}]}

Input: The church in Australia said on Monday
    Lynch , Batchelor , Barton and Riel were
    held in a prison until the weekend , when
    they were moved to join the other captives
    at the compound .
Output: {"named entities": [{"name": "Australia",
     "type": "LOC"}, {"name": "Lynch", "type": "
    PER"}, {"name": "Batchelor", "type": "PER"},
     {"name": "Barton", "type": "PER"}, {"name":
     "Riel", "type": "LOC"}]}

Input: In a telephone call to a local newspaper
    from his holiday home in Spain , Dalglish
    said : " We came to the same opinion ,
    albeit the club came to it a little bit
    earlier than me . "
Output: {"named entities": [{"name": "Spain", "
    type": "LOC"}, {"name": "Dalglish", "type":
    "PER"}]}

Input: Bosnian refugees in Hungary , the first
    to vote last weekend in their country 's
    first post-war election , found the rules
    confusing and some had no idea who they
    voted for , refugees and officials said on
    Wednesday .
Output: {"named entities": [{"name": "Bosnian",
    "type": "MISC"}, {"name": "Hungary", "type":
     "LOC"}]}

Input: Glasgow Rangers striker Ally McCoist ,
    another man in form after two hat-tricks in
    four days , was also named for the August 31
     World Cup qualifier against Austria in
    Vienna .
Output: {"named entities": [{"name": "Glasgow
    Rangers", "type": "ORG"}, {"name": "Ally
    McCoist", "type": "PER"}, {"name": "World
    Cup", "type": "MISC"}, {"name": "Austria", "
    type": "LOC"}, {"name": "Vienna", "type": "
    LOC"}]}

Input: Austrian television said the coach ,
    which was carrying 45 , was en route from
    the Czech Republic to Italy when the
    accident occurred near Steinberg , 200 km
    southwest of Vienna .
Output: {"named entities": [{"name": "Austrian",
     "type": "MISC"}, {"name": "Czech Republic",
     "type": "LOC"}, {"name": "Italy", "type": "
    LOC"}, {"name": "Steinberg", "type": "LOC"},
     {"name": "Vienna", "type": "LOC"}]}

Input: Austrian television reported earlier that
     more than 20 had been hurt in the accident
    at the station in Linz , 300 km ( 180 miles
    ) west of Vienna .
Output: {"named entities": [{"name": "Austrian",
     "type": "MISC"}, {"name": "Linz", "type": "
    LOC"}, {"name": "Vienna", "type": "LOC"}]}

Input: The fans , in Austria to watch their team
     play Rapid Vienna last Wednesday , may have
     been involved in a pub brawl earlier , the
    spokeswoman said .
Output:
```

In-Context Learning - Output

```
{"named entities": [{"name": "Austria", "type":
    "LOC"}, {"name": "Rapid Vienna", "type": "
    ORG"}]}
```

Reflect Unseen - Input

```
<input_text>
Bitar pulled off fine saves whenever they did .
</input_text>

<candidate_tokens>
['Bitar']
</candidate_tokens>

<candidate_token>
Bitar
</candidate_token>
<potential_context_tokens_around>
['pulled']
</potential_context_tokens_around>
<context_token>
pulled
</context_token>
<examples>
Input: 10th-ranked American Chanda Rubin has
    pulled
Output: {"name": "Chanda Rubin", "type": "PER"}
</examples>

Given the input text from a Reuters news article,
     evaluate each candidate token along with
     the surrounding tokens to determine if it
     should be categorized as (part of) one of
     the four named entity types: "PER" (Person),
      "LOC" (Location), "ORG" (Organization), or
     "MISC" (Miscellaneous). **Use provided
     examples, if available, for reference.** If
     it should be a new entity, extract the exact
      text span in the sentence, including any
     spaces, in JSON format. Ensure tokens are
     not split (e.g., maintain hyphenated words
     intact). Note that abbreviations and full
     names are separate entities. If no changes
     are made, return {"named entities": []}.

JSON Template:
{"named entities": [{"name": "ent_name_1", "type
     ": "ent_type_1"}, ..., {"name": "ent_name_n
     ", "type": "ent_type_n"}]}

Output Format:

Candidate Token: token_1
Contextual Meaning: ...
Relation to Examples Provided: ...
Rationale: ...
Updates: ... (add a new entity/no change)

Candidate Token: token_n
Contextual Meaning: ...
Relation to Examples Provided: ...
Rationale: ...
Updates: ... (add a new entity/no change)

Final predicted entities for the input text (
     JSON format):
```

Reflect Unseen - Output

```
Candidate Token: Bitar
Contextual Meaning: "Bitar" appears to be a
     proper noun, likely referring to a person
     involved in a sporting context, as indicated
      by the phrase "pulled off fine saves."
Relation to Examples Provided: The example
     provided features a person (Chanda Rubin)
     and categorizes her as "PER." This aligns
     with the context of "Bitar," suggesting it
     is also a person's name.
Rationale: Given that "Bitar" is a proper noun
     and is likely a person's name in a sports
     context, it should be categorized as a
     person entity.
Updates: Add a new entity.

Final predicted entities for the input text (
     JSON format):
{"named entities": [{"name": "Bitar", "type": "
     PER"}]}
```

Reflect False Negative - Input

```
<input_text>
Italian Serie A games to be played on Sunday (
     league positions in parentheses , all kick-
     off times GMT ) :
</input_text>

<candidate_tokens>
['Italian']
</candidate_tokens>

<candidate_token>
Italian
</candidate_token>
<token_stat>
{"num_occurrences_as_entity": 35, "
     num_occurrences_as_context_tokens": 0, "
     num_occurrences_as_other_tokens": 0, "
     entity_vs_context_count": "35 vs 0", "
     entity_vs_non_entity_count": "35 vs 0"}
</token_stat>
<examples>
Positive Examples (part of entity):
Input: Middlesbrough 's Italian striker Fabrizio
Output: {"name": "Italian", "type": "MISC"}

Hard Negative Examples (context tokens):


Negative Examples (other tokens, not entity nor
     context):

</examples>

Please follow the instructions below:
1. Evaluate each candidate token listed above to
      determine if it should be categorized as (
     part of) one of the four named entity types:
      "PER" (Person), "LOC" (Location), "ORG" (
     Organization), or "MISC" (Miscellaneous).
     Consider both the positive and negative
     examples provided carefully. Pay particular
     attention to the overall statistical data on
      whether tokens are included or excluded
     from the entity. Hard negative examples
     highlight tokens that are not part of the
     entity but are located near it.
2. Review both sets carefully. In many cases, a
     token may be identified as part of an entity
      in positive examples but not in negative
     ones, likely due to inconsistencies in the
     annotation process. if positive and (hard)
     negative examples seem similar, **base your
     decision on statistical data**, such as the
     frequency of the span being recognized as an
      entity versus its context, particularly
     when the data is clear-cut (e.g., one
     frequency is significantly higher).
3. If the token has not been seen or is rarely
     seen in the training data, use your best
     judgment to determine whether it should be
     considered as part of or the entire name of
     a **specific** entity.
4. If any modifications are necessary, provide
     the updated entities by **extracting the
     exact text span in the sentence**, including
      any spaces and no outside tokens added, in
     JSON format. Ensure tokens are not split (e.
     g., maintain hyphenated words intact). Note
     that abbreviations and full names are
     separate entities. If no changes are
     required, return {"named entities": []}.

JSON Template:
{"named entities": [{"name": "ent_name_1", "type
     ": "ent_type_1"}, ..., {"name": "ent_name_n
     ", "type": "ent_type_n"}]}

Output Format:

Candidate Token: token_1
```

```
Training Data Stats: ... entity ... context ... regular ...
Contextual Meaning: ...
Relation to Examples Provided: ... positive examples ... negative examples ...
Rationale: ...
Updates: ... (add a new entity/no change)

Candidate Token: token_n
Training Data Stats: ... entity ... context ... regular ...
Contextual Meaning: ...
Relation to Examples Provided: ... positive examples ... negative examples ...
Rationale: ...
Updates: ... (add a new entity/no change)

Final predicted entities for the input text (JSON format):
```

Reflect False Negative - Output

```
Candidate Token: Italian
Training Data Stats: {"num_occurrences_as_entity": 35, "num_occurrences_as_context_tokens": 0, "num_occurrences_as_other_tokens": 0, "entity_vs_context_count": "35 vs 0", "entity_vs_non_entity_count": "35 vs 0"}
Contextual Meaning: Refers to something related to Italy, in this case, the Italian Serie A football league.
Relation to Examples Provided: Positive examples indicate that "Italian" is recognized as part of an entity (MISC) in contexts related to football or culture. There are no hard negative examples that contradict this.
Rationale: The statistical data strongly supports that "Italian" is consistently recognized as an entity, specifically as a miscellaneous term related to the context of sports.
Updates: No change

Final predicted entities for the input text (JSON format):
{"named entities": [{"name": "Italian", "type": "MISC"}]}
```

Reflect Boundary - Input

```
<input_text>
Xinhua did not say when Qinglan port in Wenchang city would be opened to foreign vessels .
</input_text>

<predicted_entity>
{"name": "Wenchang city", "type": "LOC"}
</predicted_entity>

<boundary_tokens>
['city']
</boundary_tokens>

<boundary_token>
city
</boundary_token>
<status>
part of the entity
</status>
<token_stat>
{"num_occurrences_as_entity": 0, "num_occurrences_as_context_tokens": 44, "num_occurrences_as_other_tokens": 20, "entity_vs_context_count": "0 vs 44", "entity_vs_non_entity_count": "0 vs 64"}
</token_stat>
<examples>
Positive Examples (part of entity):


Hard Negative Examples (context tokens):
Input: in the English city of
Output: {"name": "English", "type": "MISC"}

Negative Examples (regular tokens, neither entity nor context):
Input: said the city council would
Output: {}
</examples>

Please follow the instructions below:
1. Calibrate the boundary of the predicted entity by evaluating each boundary token listed above in relation to the predicted entity. Don't consider it if it belongs to adjacent entities. Check both the provided positive and negative examples, with particular attention to the context surrounding the boundary token and overall statistical data on inclusion or exclusion from the entity. Hard negative examples highlight tokens that are not part of the entity but are located near it.
2. Review both sets carefully. In many cases, a token may be identified as part of an entity in positive examples but not in negative ones, likely due to inconsistencies in the annotation process. if positive and (hard) negative examples seem similar, **base your decision on statistical data**, such as the frequency of the span being recognized as an entity versus its context, particularly when the data is clear-cut (e.g., one frequency is significantly higher).
3. If the token has not been seen or is rarely seen in the training data, use your best judgment to determine whether it should be considered as part of or the entire name of a **specific** entity.
4. Determine whether any modifications, such as adding or removing boundary tokens, are necessary. If changes are required, provide the updated entity by **extracting the exact text span in the sentence**, including any spaces and no outside tokens added, in JSON format. Note that abbreviations and full names are separate entities. If no tokens are added to or removed from the predicted entity, output the original entity. If all original tokens are removed, output {}.

JSON Template:
{"name": "ent_name", "type": "ent_type"}

Output Format:

Boundary Token: token_1
Training Data Stats: ... entity ... context ... regular ...
Contextual Meaning: ...
Rationale: ... positive examples ... negative examples ... data stats ...
```

```
Boundary Token: token_n
Training Data Stats: ... entity ... context ...
     regular ...
Contextual Meaning: ...
Rationale: ... positive examples ... negative
     examples ... data stats ...

Updated Predicted Entity (JSON format):
```

Reflect Boundary - Output

```
Boundary Token: city
Training Data Stats: 0 entity, 44 context, 20
     regular
Contextual Meaning: The term "city" is used to
     describe a type of location but does not
     specify a unique entity on its own.
Rationale: In the provided examples, "city"
     appears in contexts that do not contribute
     to identifying a specific entity. The
     positive examples do not include "city" as
     part of the entity, while the hard negative
     examples show that "city" is used in a
     general context. The statistical data
     indicates that "city" has not been
     recognized as part of an entity in the
     training data.

Updated Predicted Entity (JSON format):
{"name": "Wenchang", "type": "LOC"}
```

	KATE	DEER			
	ICL	ICL	Error Reflection		
Dataset	Emb.	Token	Unseen	FN	Bound.
NCBI-disease	80.0	82.7	83.0	83.0	84.8
bc2gm	73.6	74.3	74.3	74.4	75.4
CoNLL03	92.9	93.5	93.5	93.5	93.6
OntoNotes	80.7	83.5	83.4	83.6	85.2
TweetNER7	62.2	63.6	63.8	63.8	64.3

Table 8: Performance breakdown of DEER (with GPT-4o) by each step in its pipeline. The subsequent reflection steps demonstrate performance gains, validating the effectiveness of the proposed reflection strategy.

The hyper-parameters used for DEER are presented in Table 10, and the hyper-parameters used for fine-tuning BERT-base are presented in Table 11.

A.2 Statistical Test on Sampled Datasets

To ensure the sampled subset provides a reliable estimate of model performance, we calculate 95% confidence intervals for DEER's F1 scores using 1,000 bootstrap resamples. The margins of error for the three datasets are shown in Table 13 and fall within a 0.01-0.03 range, suggesting that the random sampling does not significantly affect the validity of our performance estimates.

			Unseen Ent.	
Dataset	Context	**Seen Ent.**	Seen Tok.	Unseen Tok.
NCBI-disease	0	90.4	59.8	70.5
	1	90.4	62.1	73.3
	2	90.6	64.1	**76.8**
	3	**90.8**	**65.4**	73.8
CoNLL03	0	95.5	84.9	91.2
	1	95.6	84.8	91.7
	2	**95.7**	**86.9**	92.0
	3	95.2	86.2	**92.1**
OntoNotes	0	88.1	63.7	76.2
	1	**89.5**	67.5	**80.2**
	2	88.6	67.8	79.6
	3	89.4	**68.3**	78.9

Table 9: Performance breakdown of DEER on seen and unseen entities with varying context lengths (where 0 indicates the removal of context tokens). The results demonstrate that context tokens significantly enhance DEER's performance, particularly on unseen entities.

Dataset	λ_1	λ_2	C	w_e	w_c	w_o	θ_{FN}	M	K
NCBI	1	1	2	1.0	1.0	0.01	0.95	1	2
bc2gm	1	0.01	2	1.0	0.5	0.01	0.9	1	2
CoNLL03	0.01	1	2	1.0	1.0	0.01	0.95	1	2
OntoNotes	1	1	2	1.0	1.0	0.01	0.95	4	2
TweetNER7	1	1	2	1.0	1.0	0.01	0.95	1	2

Table 10: Hyper-parameters for DEER. λ_1 and λ_2 control the relative contribution of two components in the retrieval. C refers to context length. w_e, w_c, and w_o weight the relative importance of three token types. θ_{FN} is set as a threshold for reflections on false negative tokens. M is the number of span-level demonstrations in the reflection prompt. K is the number of tokens for boundary reflection.

learning rate	1×10^{-5}
# epoch	10
batch size	16
gradient accumulation steps	2

Table 11: Hyper-parameters for BERT-base fine-tuning.

Dataset	Unseen	FN	Bound.
NCBI	92	20	130
bc2gm	23	4	163
CoNLL03	38	12	29
OntoNotes	35	21	115
TweetNER7	174	6	215

Table 12: The number of reflections triggered across five datasets using GPT-4o. The majority of reflections are concentrated on boundary tokens, while the number of reflections involving unseen tokens remains relatively small and computationally manageable.

B Ablation Studies

In addition to the ablation studies presented in §4.1, we further investigate the impact of other compo-

Dataset	Error Margin
bc2gm	0.027
CoNLL03	0.013
OntoNotes	0.019

Table 13: Margins of error for three sampled datasets.

Method	$\text{BERT}_{\text{base}}$	$\text{Qwen2.5}_{\text{7B}}$	`GPT-4o-mini`	`GPT-4o`
Fine-tune	84.0	-	-	-
KATE	-	66.9	76.0	80.0
GPT-NER	-	71.2	75.6	82.7
DEER (w/o ER)	-	**71.7**	**77.6**	**82.7**

Table 14: Comparing our DEER with GPT-NER. DEER consistently outperforms GPT-NER across different LLMs. Notably, GPT-NER even underperforms its original BERT-based retriever, highlighting its limitations and impracticality.

nents of DEER below.

		Tagging	Python func.	JSON
Dataset	LLM	(Paolini et al., 2021)	(Li et al., 2023)	(ours)
NCBI	`GPT-4o-mini`	71.3	74.8	77.6
	`GPT-4o`	80.1	81.7	82.7

Table 15: Comparing our JSON output format with prompting formats from recent works (Paolini et al., 2021; Li et al., 2023) for the ICL step in DEER, using `GPT-4o-mini` and `GPT-4o`. Our JSON format consistently outperforms the other two prompting formats, though the performance gap narrows when using `GPT-4o`.

B.1 Prompting Template

We compare our JSON output format (Dunn et al., 2022; Lu et al., 2022; Bai et al., 2024) with other prompting formats explored in previous studies, including tagging-based formats (Paolini et al., 2021; Wang et al., 2023a), such as *"[Barack Obama | PERSON] was ..."*, and Python function generation (Li et al., 2023). Results in Table 15 show that our selected format outperforms two other formats, though the performance gap narrows when using `GPT-4o`.

B.2 Retrieval Embeddings

Following previous work (Monajatipoor et al., 2024), we experiment with seven embedding models to identify the most suitable sentence embeddings for KATE. These models include six open-source models available on HuggingFace[7] via the `sentence-transformer` library[8], along with OpenAI's proprietary model `text-embedding-3-small`. For open-source models, we include both general-domain models, such as `all-mpnet-base-v2`, and domain-specific models, such as `BioClinicalBERT`. Table 16 presents the performance of these seven embedding models with KATE using `GPT-4o-mini`. The results show that `BioClinicalBERT` outperforms other open-source models, highlighting the benefits of in-domain pre-training. Among all models, `text-embedding-3-small` achieves the best performance, surpassing others by a clear margin. Consequently, we adopt `text-embedding-3-small` for all datasets in our experiments.

Emb. Model	KATE
`bert-base-cased`	70.9
`bert-base-uncased`	72.8
`all-MiniLM-L6-v2`	73.2
`all-mpnet-base-v2`	73.3
`BioBERT`	72.2
`BioClinicalBERT`	73.6
`text-embedding-3-small`	**76.0**

Table 16: Comparing seven embedding models to select the most suitable sentence embeddings for KATE using `GPT-4o-mini`. `text-embedding-3-small` surpasses other models by a clear margin, thus adopted for all our experiments.

B.3 Contextualized v.s. Uncontextualized Embeddings

We compare two types of token embeddings for our token-focused retriever. Contextualized token embeddings are obtained by averaging subtoken embeddings from the last layer of the embedding model, capturing sentence-level contextual information. Uncontextualized embeddings encode each token in the vocabulary independently. We evaluate both types using seven embedding models. Note that OpenAI's `text-embedding-3-small` does not provide sub-token embeddings from input text, so we use it only for uncontextualized token embeddings. The generated token embeddings are then weighted by training set token-level statistics to produce sentence-level embeddings for demonstration retrieval, as introduced in §2. The results show no consistent pattern among open-source models regarding which type of token embeddings performs best; the outcome varies

[7] `https://huggingface.co/models`

[8] `https://sbert.net/`

depending on the model. However, among all models, the uncontextualized embeddings from `text-embedding-3-small` achieve the best performance.

Emb. Model	Contex.	Uncontex.
`bert-base-cased`	74.9	74.9
`bert-base-uncased`	73.9	75.9
`all-MiniLM-L6-v2`	74.5	75.1
`all-mpnet-base-v2`	73.4	74.7
`BioBERT`	74.1	74.6
`BioClinicalBERT`	75.9	74.7
`text-embedding-3-small`	-	**76.3**

Table 17: Comparing contextualized and uncontextualized token embeddings used in DEER on NCBI. No consistent pattern among open-source models is found regarding which type of token embeddings performs best. Uncontextualized embeddings from `text-embedding-3-small` achieve the best performance.

B.4 Two Components in Label-guided Retriever

We compare four retrieval methods: sentence embeddings used in KATE (`S-Emb`), our token-based retriever, and its two components, weighted token matching (`T-Match`) and weighted token embeddings (`T-Emb`). The results across three datasets are presented in Table 18. The results show that `T-Emb` consistently outperforms `S-Emb`, underscoring the importance of prioritizing entity- and context-related tokens in demonstration retrieval for NER. Combining `T-Emb` with `T-Match` improves performance, though `T-Match` alone lags behind `S-Emb` on CoNLL03. Further analysis reveals that this issue stems from short, non-language data in the dataset, such as news bylines (e.g., "OMAHA 1996-12-06") or sports score tables, as noted by Liu and Ritter (2023). These examples contain a high proportion of unseen tokens, reducing the effectiveness of `T-Match`, which relies on exact token matching.

B.5 Token Type Weights

As explained in §2, we use hyperparameters w_e, w_c, and w_o to control the relative importance of the three token types. Their values are determined through a grid search over three predefined values: $[1.0, 0.5, 0.01]$. Due to budget constraints, we select $w_e = 1.0$, $w_c = 1.0$, and $w_o = 0.01$ based on initial trials. This choice aligns with the intuition that entity- and context-related tokens are more important for NER. We find that these values perform well across most datasets. To evaluate the importance of each token type, we examine three sets of values, assigning 0.01 to each weight in turn while keeping the others at 1.0. Table 19 presents the results of the ICL step in DEER under these three settings. The results show that the default set $w_e = 1.0$, $w_c = 1.0$, and $w_o = 0.01$ performs best, while the other two sets exhibit significant performance drops. This outcome verifies the critical importance of entity- and context-related tokens for NER performance.

	KATE	DEER		
Retriever	S-Emb	T-Match	T-Emb	Combined
NCBI	80.0	82.0	81.4	**82.7**
CoNLL03	92.9	92.2	93.3	**93.5**
OntoNotes	80.7	82.6	82.3	**83.5**

Table 18: Comparison of four demonstration retrievers: sentence embeddings (`S-Emb`), two components of DEER (`T-Emb` and `T-Match`), and their combination. `T-Emb` consistently outperforms `S-Emb`, highlighting the importance of prioritizing entity- and context-related tokens in retrieval. Combining `T-Emb` with `T-Match` further enhances performance.

w_e	w_c	w_o	DEER
0.01	1.0	1.0	71.6
1.0	0.01	1.0	74.7
1.0	1.0	0.01	**77.6**

Table 19: Comparing three sets of values, assigning 0.01 to each weight in turn while keeping the others at 1.0, to evaluate the importance of each token type. The default set $w_e = 1.0$, $w_c = 1.0$, and $w_o = 0.01$ performs best, verifying the importance of entity- and context-related tokens for NER performance.

B.6 Llama3.1-8B Inadequacy

In early experiments with `Llama3.1-8B`, we observed frequent failures to generate outputs in the required JSON format during ICL inference. Table 21 reports the percentage of test examples with format errors across five datasets. The failure rates range from 26.8% to 82.7%, making the model's outputs unreliable and difficult to evaluate. Due to these inconsistencies, we exclude `Llama3.1-8B` from our main experiments.

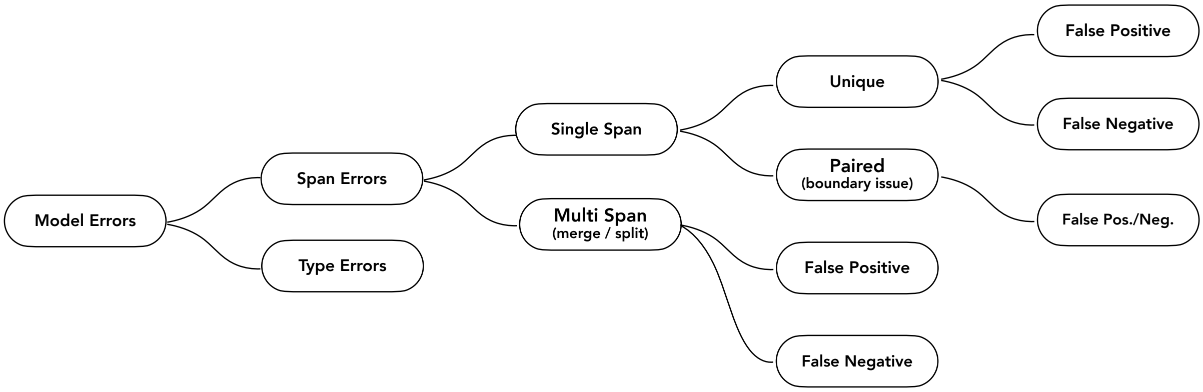

Figure 4: NER error ontology. This ontology is developed based on the standard strict mention-level matching metric for NER (Segura-Bedmar et al., 2013), where a predicted entity is considered correct only if both the predicted span boundaries and the entity type match with the gold ones. "Span Errors" refer to predictions that include at least span errors, while "Type Errors" denote cases where only the entity type is incorrect. "Multi-Span Errors" describe cases where multiple span issues occur, such as when a single gold entity is split into two predicted entities or when two gold entities are merged into one predicted entity. Similar analyses can be found in recent works (Ding et al., 2024; Lu et al., 2024).

	Kate (GPT-4o)						Deer (GPT-4o)						BERT-base					
	Type	Span					Type	Span					Type	Span				
		Multiple		Single				Multiple		Single				Multiple		Single		
				Unique		Paired				Unique		Paired				Unique		Paired
Dataset	FP/FN	FP	FN	FP	FN	FP/FN	FP/FN	FP	FN	FP	FN	FP/FN	FP/FN	FP	FN	FP	FN	FP/FN
NCBI (2014)	0	15	8	57	121	87	0	14	8	90	39	73	0	18	11	90	53	72
bc2gm (2008)	0	48	77	90	115	160	0	59	89	154	48	130	0	98	75	103	63	97
CONLL03 (2003)	53	4	8	31	44	21	54	3	6	40	23	23	66	17	16	50	28	38
OntoNotes (2013)	33	27	30	46	73	130	30	31	44	54	27	158	27	43	30	55	44	61
TweetNER7 (2022)	210	64	44	191	394	123	206	66	50	195	364	120	206	136	71	277	355	135

Table 20: Error breakdown for Kate, Deer, and BERT-base based on the hierarchical ontology of NER errors (see Figure 4). The primary source of errors across all five datasets is span-related issues for single entities. Deer, as intended by its design, performs better in reducing unique false negatives and paired errors compared to Kate. On the two datasets where Deer lags behind BERT-base, the latter still exhibits superior boundary handling, suggesting the need for a better approach to further improve span prediction.

Dataset	Failure Rate (%)
NCBI	30.7
bc2gm	26.8
CoNLL03	81.3
OntoNotes	82.7
TweetNER7	67.5

Table 21: Percentage of example outputs from `Llama3.1-8B` that failed to follow the required JSON format specified in the ICL prompt.

C Error Analyses

We conduct an error breakdown based on the hierarchical ontology of NER errors in Figure 4 for Kate, Deer, and BERT-base. The full results are presented in Table 20, revealing that the primary source of errors across all five datasets is span-related issues for single entities, as highlighted in §2 when motivating the error reflection process. These errors consist of unique false positives and negatives, as well as paired errors, typically caused by boundary mismatches. Deer outperforms Kate in reducing unique false negatives and paired errors, as intended by its design.