acrt AI, Computer Science and Robotics Technology

IntechOpen Journals

RESEARCH PAPER

Large Language Models for Few-Shot Named Entity Recognition

Yufei Zhao[1], Xiaoshi Zhong[1,*], Erik Cambria[2] and Jagath C. Rajapakse[2]

1 School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
2 College of Computing and Data Science, Nanyang Technological University, Singapore, Singapore
*Corresponding author. E-mail: xszhong@bit.edu.cn

Abstract

Named entity recognition (NER) is a fundamental task in numerous downstream applications. Recently, researchers have employed pre-trained language models (PLMs) and large language models (LLMs) to address this task. However, fully leveraging the capabilities of PLMs and LLMs with minimal human effort remains challenging. In this article, we propose GPT4NER, a method that prompts LLMs to resolve the few-shot NER task. GPT4NER constructs effective prompts using three key components: entity definition, few-shot examples, and chain-of-thought. By prompting LLMs with these effective prompts, GPT4NER transforms few-shot NER, traditionally considered as a sequence labeling problem, into a sequence-generation problem. We conduct experiments on two benchmark datasets, CoNLL2003 and OntoNotes5.0, and compare the performance of GPT4NER to representative state-of-the-art models in both few-shot and fully supervised settings. Experimental results demonstrate that GPT4NER achieves an F_1 score of 83.15% on CoNLL2003 and 70.37% on OntoNotes5.0, significantly outperforming few-shot baselines by an average margin of 7 points. Compared to fully supervised baselines, GPT4NER achieves 87.9% of its best performance on CoNLL2003 and 76.4% of its best performance on OntoNotes5.0. We also utilize a relaxed-match metric for evaluation and report performance in the sub-task of named entity extraction (NEE), and experiments demonstrate their usefulness to help better understand model behaviors in the NER task.

Keywords: few-shot, large language models, named entity recognition, prompt

Citation
Yufei Zhao, Xiaoshi Zhong, Erik Cambria and Jagath C. Rajapakse (2025), Large Language Models for Few-Shot Named Entity Recognition. *AI, Computer Science and Robotics Technology* 4, 1–31.

DOI
https://doi.org/10.5772/acrt.20250103

Received: 13 August 2025
Accepted: 28 October 2025
Published: 13 November 2025

1. Introduction

Named entity recognition (NER) [1] is a fundamental task in natural language processing, aiming to extract and classify named entities from unstructured text. By transforming original text into structured data, NER provides crucial support for many downstream tasks, making its accuracy essential for subsequent tasks. Traditionally, NER is treated as a sequence-labeling task [2, 3]. Early methods primarily relied on large annotated corpora from specific domains and employed supervised or semi-supervised learning algorithms to address this task [3, 4]. While these methods perform well on closed datasets [5, 6], they often require access to complete labeled training datasets for model training and fail to meet the demands of open-ended business scenarios in industry due to limitations in labeled data and the scarcity of data in specific domains such as biomedicine and materials science.

In the early stages, the NER task [1] primarily relied on rule-based methods [7, 8] and dictionary-based methods [9, 10], which required experts to manually construct rules based on dataset features. This process is both time-consuming and labor-intensive. With the advancement of machine-learning techniques, researchers adopt machine learning-based methods to resolve the NER task. Hidden Markov models [10–12] and conditional random fields [13, 14] become particularly representative of this approach. While these statistical machine learning-based NER models significantly improve the performance, they require extensive manual annotation of domain-specific data, limiting their scalability and practical application. The rise of deep learning and neural network techniques further transforms the NER task. Researchers employ these methods, with commonly used NER models including convolutional neural networks [15–17] and recurrent neural networks [18, 19], among others. These deep-learning models can automatically learn feature representations from large-scale data and have achieved significant improvements in the NER task.

Devlin et al. [2] transfer the pre-trained language model BERT to fine-tuning on 11 natural language processing benchmark tasks, achieving state-of-the-art results. Since then, NER methods have increasingly relied on large-scale pre-trained language models, which leverage the benefits of big data and large-scale computing. Wang et al. [20] propose structural pre-training, which guides language models to generate structures from text and enhances knowledge transfer between different tasks. Context learning has also been applied to the NER task. For example, Chen et al. [21] design a meta-function pre-training algorithm to inject context learning capabilities into pre-trained language models, enabling rapid identification of new entity types using demonstration instances. Additionally, data augmentation techniques have been used to alleviate the scarcity of labeled data in NER. For example, Hu et al. [22] propose an entity-to-text data augmentation technique that utilizes pre-trained large-scale language models to construct an augmented entity list.

With the rise and widespread use of large language models (LLMs) such as OpenAI's GPT series (e.g., GPT-3 [23] and GPT-4 [24]), various NLP tasks have achieved promising results, including relation extraction [25, 26] and question answering [27]. Trained on diverse datasets across multiple domains, LLMs exhibit powerful capabilities in understanding context and generating natural language text. With only a few examples as demonstrations for a specific task, LLMs can generate accurate responses to new inputs. In the era of LLMs, numerous studies have explored their application to NER tasks, including few-shot learning [28–30], zero-shot learning [31–33], fine-tuning models for target domains, and using GPT as a data generator for data augmentation [34, 35]. Zhao et al. [36] propose a few-shot biomedical NER method that combines LLM-assisted data augmentation with multi-scale feature extraction, effectively improving model performance on multiple biomedical datasets under few-shot settings. Few-shot methods typically include domain transfer and prompt engineering. Domain transfer methods [37, 38] usually involve training on large amounts of source data and fine-tuning on examples from the target domain. Prompt engineering methods [29, 39, 40] often adopt a strategy of querying for the presence of one specific entity type at a time to improve recognition accuracy. However, this querying approach significantly increases processing time when dealing with multiple entity types, especially when handling more entity types or longer test texts. The time cost becomes a critical bottleneck in such cases.

Chain-of-thought (CoT) prompting provides statement reasoning, maintaining complete interpretability [41]. However, it performs poorly when addressing problems more complex than the provided examples. Zhou et al. [42] introduce "least-to-most prompting," which decomposes a complex problem into a series of sub-problems and addresses them sequentially, enabling the model to solve problems harder than the examples. Zhang et al. [43] propose the auto-CoT paradigm, which automatically constructs questions and reasoning chains, improving fault tolerance.

Ashok et al. [30] apply CoT prompting to few-shot NER tasks and achieve cross-domain applications and improve flexibility by modifying definitions and examples. However, the few-shot examples are selected randomly without a targeted selection strategy, and evaluation is conducted on a random sample of 500 test examples by reporting mean and variance over 5 runs, which may not reflect performance across the full dataset or multi-type entity scenarios. Wang et al. [29] apply GPT-3 to the NER task by converting sequence labeling into a generation task, requiring the identification of entity types after providing prompts and examples (obtaining the nearest neighbors as examples through k-nearest neighbors). They use a self-verification strategy to address the hallucination problem of LLMs. However, this method can only extract one type of entity at a time. In datasets with many entity types, this can result in more time spent. Zhou et al. [39] compare querying all types of entities at once to

querying one type of entity at a time, concluding that the former mode is not efficient. Guo et al. [44] propose BANER, a boundary-aware NER framework leveraging contrastive learning and LoRAHub for cross-domain adaptation. While BANER improves entity boundary detection in few-shot settings, it employs a single, stage-specific prompt template for each phase, which may limit flexibility and expressiveness.

Inspired by LLM-based methods such as PromptNER [30], GPT-NER [29], and BANER [44], in this article, we propose GPT4NER, an LLM-based method that leverages the capabilities of LLMs to tackle the few-shot NER task. GPT4NER enables querying for all entity types in a single query, reducing querying time. GPT4NER constructs effective prompts using three key components: entity definition, few-shot examples, and CoT, with an optional component of part-of-speech (POS) tags. The entity definition component provides detailed definitions and identification criteria for each entity type in the dataset, including boundary delineation and clarification of classification confusion points. We design a selection procedure to choose few-shot examples that cover all entity types and those difficult entities to generate, implicitly specifying the output format. We sample the training data during the few-shot examples construction process, rather than using all the training data. The CoT component guides LLMs to provide reasoning for their output, enhancing the quality of generation. POS tags optionally supply syntactic information for contextual text. These components ensure that the prompts embody clear instructions, task-relevant background, and a defined output format, facilitating optimal model comprehension for the few-shot NER task.

GPT4NER differs from previous LLM-based NER methods in several aspects. Unlike PromptNER [30], which randomly selects few-shot examples, GPT4NER implements a targeted selection strategy that ensures coverage of all entity types and difficult-to-generate entities, which improves reliability across multi-type scenarios. Compared to GPT-NER [29], which queries one entity type at a time and requires a separate verification strategy to handle hallucinations, GPT4NER can query all entity types in a single call. Compared to BANER [44], which decomposes NER into a two-stage process and employs stage-specific prompt templates primarily focused on boundary detection, GPT4NER performs end-to-end entity recognition with prompts combining entity definitions, few-shot examples, CoT reasoning, and optional POS tags, allowing expressive instructions and structured output guidance.

To evaluate the effectiveness of GPT4NER, we conduct experiments on two benchmark datasets, CoNLL2003 [45] and OntoNotes5.0 [46], focusing on the few-shot NER task and its sub-task of named entity extraction (NEE). We compare the results of GPT4NER to two types of representative state-of-the-art models: few-shot models and fully supervised models. Experimental results demonstrate that GPT4NER achieves an F_1 score of 83.15% on CoNLL2003 and 70.37% on OntoNotes5.0, significantly outperforming few-shot baselines by an

average margin of 7 points. Compared to fully supervised baselines, GPT4NER achieves 87.9% of its best performance on CoNLL2003 and 76.4% of its best performance on OntoNotes5.0. Furthermore, our experiments utilize the relaxed-match metric, which is widely used for evaluating time expression recognition and normalization [47–53], to evaluate the performance of few-shot models. Our analysis indicates that while few-shot models may not precisely recognize or generate the boundaries of named entities, they can identify or generate portions of these entities. Additionally, our findings highlight the importance of reporting model performance in the sub-task of few-shot NEE to better understand model capabilities in the few-shot NER task.

In summary, the main contributions of this article are as follows:

- We propose GPT4NER, a method that prompts LLMs for few-shot NER. GPT4NER constructs effective prompts using three key components: entity definition, few-shot examples, and CoT, along with one optional component, POS tags, and adopts a targeted selection strategy that ensures coverage of all entity types and difficult-to-generate entities.
- We conduct experiments on two benchmark datasets, and the experimental results demonstrate that GPT4NER significantly outperforms representative state-of-the-art few-shot models and achieves approximately 82.15% of the best performance of fully supervised models.
- Our experiments suggest that utilizing a relaxed-match metric for evaluation can enhance our understanding of model capabilities and that reporting NEE performance provides further insights into model capabilities in the NER task.

2. *Methodology*

Figure 1 provides an overview of GPT4NER for few-shot NER, which comprises two parts: (1) prompt construction and (2) entity generation by LLMs. The prompt is built from three core elements: entity definitions, few-shot examples with CoT reasoning, and the input test text.

2.1. Prompt Construction

In leveraging the capabilities of LLMs, constructing effective prompts is crucial, laying the foundation for subsequent model training and inference processes. An exemplary prompt should embody the following three key characteristics to ensure optimal model comprehension and performance:

- **Clear Instructions**: An effective prompt must provide explicit and unambiguous instructions regarding the objectives and requirements of the task. Such clarity is essential to facilitate an accurate understanding of the core content.

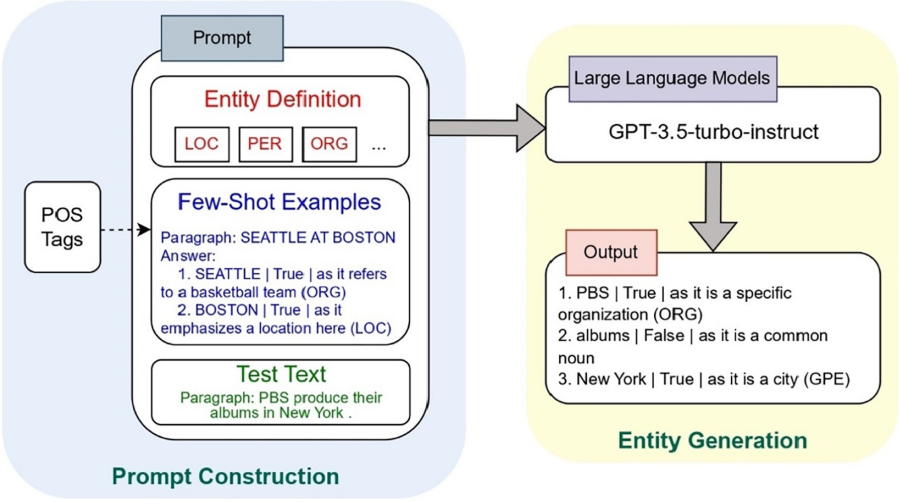

Figure 1. Overview of GPT4NER for few-shot NER. The left-hand side illustrates the prompt construction using three kinds of information: (1) entity definition, (2) few-shot examples with chain-of-thought reasoning, and (3) input test text. The right-hand side depicts the procedure of LLMs processing prompts and generating entities.

- **Task-Relevant Background**: An effective prompt should integrate task-relevant background knowledge, encompassing domain-specific expertise, entity attributes, or several examples.
- **Output Format**: An effective prompt should specify the output format, either explicitly or implicitly, because the output format directly impacts subsequent processing and evaluation. A chaotic or disorganized output format can pose significant challenges for processing and evaluation tasks.

To embody these key characteristics, we construct effective prompts that include three key components: (1) entity definition, (2) few-shot examples with output format, and (3) CoT, along with one optional component: syntactic POS information. Below, we detail these components with an example of effective prompts designed for the CoNLL2003 dataset, as illustrated in Figure 2.

2.1.1. *Entity Definition*

In different datasets, researchers may specify significantly diverse definitions and identification rules for the same types of entities. For example, both CoNLL2003 and OntoNotes5.0 include LOC entities, but their definitions differ. CoNLL2003 classifies location (LOC) entities as countries, cities, regions, such as “London” and “Germany.” By contrast, OntoNotes5.0 defines LOC entities as non- geopolitical entity (non-GPE) locations, including mountain ranges and planets. Furthermore, OntoNotes5.0 includes GPE (i.e., geopolitical entities like countries and cities) and FAC (i.e., facilities like buildings and roads), which can overlap with LOC in some cases.

Prompt

Definition:
An entity is a real object or concept representing a person (PER), named organization (ORG), location (LOC), country (LOC), or nationality (MISC). Most entities are expressed as proper nouns (NNP or NNPS) while nationality and language entities under MISC are adjectives. Names, first names, last names, and countries are entities. Nationalities are considered entities even if they are adjectives. However, sports, sporting events, adjectives, verbs, numbers, adverbs, and abstract concepts are not considered entities. Similarly, dates, years, weekdays, months, and times are not entities. Possessive words and pronouns like "I", "you", "him", and "me" are not entities. If a sporting team is referred to by the name of their location, the term is considered an entity representing an organization, not a location.

Example:
Q: Given the paragraph below, identify a list of possible entities and for each entry explain why it either is or is not an entity.
Paragraph:
South African all - rounder Shaun Pollock , forced to cut short his first season with Warwickshire to have ankle surgery , has told the English county he would like to return later in his career .
Answer:
1. South African | True | as it is an adjective representing nationality (MISC)
2. Shaun Pollock | True | as it is a person's name (PER)
3. Warwickshire | True | as it refers to a cricket club, emphasizing an organization here (ORG)
4. English | True | as it is an adjective representing nationality (MISC)
5. county | False | as it is a common noun
6. he | False | as it is a pronoun
7. career | False | as it is a common noun

Test Text:
Q: Given the paragraph below, identify a list of possible entities and for each entry explain why it either is or is not an entity.
Paragraph:
SOCCER - JAPAN GET LUCKY WIN , CHINA IN SURPRISE DEFEAT .
Answer:

Figure 2. An example of effective prompts for the CoNLL2003 dataset. Entity definition is in red. Few-shot examples with question-answer format and CoT reason are in blue. Test text is in dark green.

Therefore, simply hinting at differences between entity types in few-shot examples may impede the performance of traditional few-shot methods. It is crucial to provide explicit and unambiguous definitions and meticulous identification rules for each entity type within the dataset. Additionally, it is important to delineate boundary conditions and elucidate points of potential classification ambiguity. For example, OntoNotes5.0 specifies PERSON to include generational markers (e.g., "Jr." and "IV") while excluding honorifics (e.g., "Ms." and "Dr.") and occupational titles (e.g., "President" and "Secretary"). Explicitly describing the scope of an entity helps precisely identify the boundaries of entities.

An inherent challenge in prompt construction is reconciling the need for domain-specific knowledge with users' limited understanding. To resolve this challenge, we express entity definitions in natural language, avoiding excessive technical jargon. Such a strategy not only facilitates comprehension but also provides greater flexibility, allowing for adaptable use across diverse datasets without sacrificing specificity.

In the entity definition module, we adhere to these principles by providing detailed definitions and identification criteria in natural language for each entity type within individual datasets. These definitions include boundary delineation and points of classification confusion. In this article, we utilize two benchmark datasets: CoNLL2003 [45] and OntoNotes5.0 [46]. For CoNLL2003, we construct the definition for its entities, as illustrated in Figure 2. For OntoNotes5.0, we construct the following definition for its entities:

An entity is a real object or concept that represents an event, facility, country, language, location, nationality, organization, person, product, or work of art. Typically, entities are expressed as proper nouns (NNP or NNPs). Event (EVENT) entities refer to proper nouns representing hurricanes, battles, wars, sports events, and attacks. Facility (FAC) entities refer to proper nouns associated with man-made structures like buildings, airports, highways, and bridges. Geographical (GPE) entities refer to proper nouns representing countries, cities, states, provinces, and municipalities. Language (LANGUAGE) entities refer to named languages. Location (LOC) entities refer to proper nouns representing non-GPE locations, including mountain ranges, planets, geo-coordinates, bodies of water, named regions, and continents. Nationalities, religious, or political groups (NORP) are expressed through adjectival forms of geographical, social, and political entities, location names, named religions, heritage, and political affiliations. Organization (ORG) entities refer to proper nouns representing companies, government agencies, educational institutions, and sports teams. These also include adjectival forms of organization names and metonymic mentions of associated buildings or locations. Person (PERSON) entities are represented by proper personal names, including fictional characters, first names, last names, nicknames, and generational markers (such as Jr. and IV), excluding occupational titles and honorifics. Product (PRODUCT) entities refer to proper nouns representing model names, vehicles, or weapons. Manufacturer and product should be marked separately. Works of art (WORK_OF_ART) refer to titles of books, songs, articles, television programs, or awards. If an organization, occupation title, and person's name form a phrase, then the organization and person's name is marked separately. Nominals and common nouns are not considered entities. Additionally, pronouns and pronominal elements are excluded from entities, as are contact information, plants, dates, years, times, numbers, legal documents, treaties, credit cards, checking accounts, CDs, credit plans, financial instruments, and abstract concepts.

2.1.2. *Few-Shot Examples with Implicit Output Format*

Few-shot examples serve as a vital instructional tool in prompts, providing tangible exemplars of contextual instantiation for each entity type. The inclusion of these examples aims to afford the model with invaluable insights into the contextual nuances underpinning entity identification.

Many LLMs have strict limitations on the maximum number of tokens they can process (e.g., OpenAI's GPT-3.5-turbo-instruct model supports only a 4K-token window). Consequently, each input can accommodate up to 10 examples, with around 400–500 tokens reserved for output.

In our constructed prompts, each example comprises three types of information: (1) *a task-description question*, (2) *an input text*, and (3) *output results*. These few-shot examples provide direct instructions and evidence relevant to the task, enabling LLMs to grasp the logic of predictions.

Algorithm 1 Optimizing Example Selection for Limited Tokens

Input: Training set $\mathcal{D}$, maximum token limit T
Output: Optimized few-shot examples $\mathcal{P}$
$\mathcal{P} \leftarrow \emptyset$
Step 1: Select Texts with Multiple Entity Types
Select texts $\mathcal{T}_1 \subseteq \mathcal{D}$ with at least 3 entity types
$\mathcal{P} \leftarrow \mathcal{P} \cup$ {Select 3-4 texts from $\mathcal{T}_1$ covering all entity types}
Step 2: Identify and Test Confused Entities
Select confused entities $\varepsilon_c \subseteq \mathcal{D}$
for each text $t \in \varepsilon_c$ **do**
Test t using current prompt $\mathcal{P}$
if entity recognition is suboptimal **then**
$\mathcal{P} \leftarrow \mathcal{P} \bigcup \{t\}$
end if
end for
Step 3: Sample and Finalize Examples
while number of examples in $\mathcal{P}$ is less than 10 and token count $< T$ **do**
Randomly sample 10 texts $\mathcal{T}_2 \subseteq \mathcal{D}$ each time
for each text $t \in \mathcal{T}_2$ **do**
Test t using current prompt $\mathcal{P}$
if entity recognition is suboptimal **then**
$\mathcal{P} \leftarrow \mathcal{P} \bigcup \{t\}$
end if
end for
end while
return $\mathcal{P}$

Task-Description Question: The task-description question serves to guide LLMs on the task at hand. We utilize the following format as the task instruction:

- *Q: Given the paragraph below, identify a list of possible entities and for each entry explain why it either is or is not an entity.*

Input Text: The input text is selected as an example from the training data, with the primary objective of enhancing the accuracy of recognizing entities in test text. The selection process prioritizes texts that closely resemble the test text, especially those presenting identification challenges. These chosen texts often exhibit more complex results and involve entity categories that are easily confused, including both positive and negative instances (i.e., examples of both entities and non-entities).

Few-shot examples are thoughtfully selected to illustrate diverse contexts in which a given entity type may manifest, encompassing variations in syntactic structure and semantic context. By exposing the model to a range of context understanding and entity recognition instances, we aim to imbue the model with a robust understanding of the myriad manifestations of entity types, thereby enhancing its adaptability and generalization capabilities. To achieve this, we select and adjust examples through multiple sampling tests based on feedback from results. These few-shot examples include challenges in identifying entities and understanding specific contexts. Given the limited number of tokens specified by LLMs, we carefully select these examples. This selection procedure mainly comprises the following three steps, as illustrated in Algorithm 1.

- **Step 1: Select texts with multiple types of entities**. From the training set, select texts containing at least three types of entities. Choose three to four texts to ensure that all entity types are covered for a 1-shot setup.
- **Step 2: Identify confused entities**. Conduct a small-scale test using the texts selected in **Step 1** to evaluate the prompt's effectiveness, and then add the texts whose entities are poorly generated as new examples to the prompt.
- **Step 3: Check for omissions**. Randomly select 10 texts for testing each time. Gradually add examples following **Step 2** until the number of added examples reaches 10, the maximum number of examples.

Output Format. The output format is implicitly incorporated into prompts alongside entity definition and few-shot examples. It delineates the expected format for the output labels corresponding to identified entities. These implicit output formats serve as guiding beacons, steering the model toward generating output labels that adhere to predefined standards of clarity, consistency, and conciseness. By embedding an intrinsic awareness of annotation conventions within the model, these implicit output formats ensure that outputs are semantically accurate and adhere to established annotation standards.

Each test sentence needs to satisfy the following conditions: (1) it needs to clearly list words or phrases that are (or are not) entities and their corresponding category labels; (2) it needs to be easy for LLMs to learn and imitate so that we can smoothly label each token in the test sentence. The output includes a list of candidate entities, explanations for identification and classification, and specific entity types. The output format is structured as follows:

Candidate | True or False | Explanation of why the candidate is or is not an entity [(Type)]

which contains three elements:

- *Candidate*: This element indicates a generated candidate that may be considered as an entity.

- *True or False*: This element indicates whether the generated candidate is an entity. Specifically, "*True*" indicates that the candidate is an entity, while "*False*" indicates that it is not.
- *Explanation of why the candidate is or is not an entity [(Type)]*: This element explains why the candidate is or is not treated as an entity and specifies the type of the entity if the candidate is an entity. It is our designed CoT component and will be described in Section 2.1.3.

For example, as shown in Figure 2, the first entry of the output is "South African | True | as it is an adjective representing nationality (MISC)." This means that "South African" is a generated candidate that may be treated as an entity. "True" indicates that "South African" is indeed treated as an entity. The explanation "as it is an adjective representing nationality (MISC)" clarifies why "South African" is treated as an entity, specifically under the type MISC. By contrast, the fifth entry, "county | False | as it is a common noun," indicates that "county" is a candidate, but "False" suggests that this candidate is not an entity. The explanation, "as it is a common noun," clarifies why "county" is not treated as an entity.

2.1.3. CoT

Incorporating explanations of whether candidates are entities into the prompt enhances the clarity of the instruction and serves as a practical implementation of our CoT reasoning. This approach strategically guides LLMs through a systematic thought process, encouraging careful consideration of each step and coherent explanations for its decisions. Recent findings suggest that CoT prompting can guide LLMs to output reasoning, even by simply adding "think step by step" to the prompt [41, 43, 54]. Numerous studies have underscored the effectiveness of this approach, showing that guiding LLMs to think step by step and articulate their reasoning can greatly reduce errors. By asking LLMs to provide a reason for recognition along with generating the entity list, we can substantially improve the reliability of the outputs. In this work, CoT explanations are incorporated into the output format of the prompt, as described in Section 2.1.2.

2.1.4. POS Information

Zhong et al. [52] demonstrate that named entities are primarily composed of proper nouns, and Ye et al. [35] show that data augmentation using LLMs to alter the syntactic structure of input text can enhance few-shot NER. Therefore, we incorporate POS tags to enrich the syntactic information of named entities in the text. Specifically, POS tags are included in these few-shot examples as part of the input text, as shown below:

South/NNP African/JJ all/DT -/HYPH rounder/NN Shaun/NNP Pollock/NNP,/, forced/VBN to/TO cut/VB short/IN his/PRP first/JJ season/NN with/IN Warwick shire/NNP to/TO have/VB ankle/NN surgery/NN,/, has/VBZ told/

VBN the/DT English/JJ county/NN he/PRP would/MD like/VB to/TO return/VB later/RBR in/IN his/PRP career/NN ./.

2.2. Entity Generation by LLMs

Recent studies approach the NER task as a sequence-to-sequence problem and employ methods such as prompt-based techniques or in-context learning. We adopt a similar perspective to address the NER task through sequence generation by prompting LLMs. A primary motivation for treating NER as a sequence-generation problem is to mitigate the challenge of combinatorial explosion, which arises when entities consist of multiple tokens. Traditional token-based approaches may struggle to handle such cases effectively, leading to suboptimal performance and decreased accuracy. By contrast, LLMs have demonstrated noteworthy performance in NER tasks, even when they are trained on only a small subset of training data. This highlights the efficacy of leveraging unsupervised pre-trained models for sequence generation tasks, where the model can generalize effectively from a limited number of examples to achieve competitive performance across diverse datasets and domains. Unlike conventional supervised models that rely heavily on labeled training data, we utilize LLMs as a powerful sequence-generation tool for few-shot NER, capitalizing on their ability to perform well with minimal labeled data.

2.3. Experimental Setup

2.3.1. Datasets

The evaluation of GPT4NER is conducted on two benchmark datasets: CoNLL2003 [45] and OntoNotes5.0 [46].

CoNLL2003 is a widely used benchmark dataset derived from the Reuters RCV1 corpus, containing 1393 news articles spanning from August 1996 to August 1997. It includes 35 089 entities categorized into four types: PER, LOC, ORG, and MISC.

Ontonotes5.0 is also a widely used benchmark dataset developed for the analysis of several linguistic tasks in three languages. In this article, we focus only on the NER task in English and use only the NER portion of the OntoNotes5.0 dataset. This subset consists of 3370 articles collected from various sources such as newswire and web data. It contains 18 types of entities, among which 10 types are primarily related to proper nouns or nationalities, while the other eight types involve changing digits. We are mainly concerned with the 10 types of concrete entities related to proper nouns or nationalities: EVENT, FAC, GPE, LANGUAGE, LOC, NORP, ORG, PERSON, PRODUCT, WORK OF ART.[1]

For CoNLL2003, we follow previous studies [17] to divide its data into training, development, and testing sets. For OntoNotes5.0, we split the data into

[1]The excluded entity types are CARDINAL, DATE, LAW, MONEY, ORDINAL, PERCENT, QUANTITY, and TIME.

Table 1. Statistics of the two benchmark datasets

Dataset		No. of Sentences	No. of Tokens	No. of Entities	No. of Types
CoNLL2003	Training set	14 987	203 621	23 499	4
	Development set	3466	51 362	5942	
	Test set	3684	46 435	5648	
OntoNotes5.0	Training set	59 924	1 088 503	55 530	10
	Development set	8528	147 724	7584	
	Test set	8262	152 728	7505	

training, development, and testing sets using the same method as Pradhan et al. [46]. Dataset statistics are summarized in Table 1.

2.3.2. *State-of-the-Art Baselines*

We compare the performance of GPT4NER to five representative state-of-the-art models, including three few-shot models and two fully supervised models.

Few-Shot Baselines:

- **ProML** [38] designs multiple prompt schemas to improve label semantics and introduces a novel architecture to combine these prompt-based representations. It targets tasks such as token set expansion and domain transfer.
- **CONTaiNER** [37] is a contrastive learning technique for few-shot NER that optimizes inter-token distribution distance using Gaussian-distributed embeddings. This method enhances differentiation between token categories and alleviates overfitting from training domains.
- **PromptNER** [30] advances entity recognition by integrating entity definitions in addition to few-shot examples and prompts language models to produce a list of potential entities along with corresponding explanations.

Fully Supervised Baselines:

- **MRC-NER + DSC** [55] employs dice loss instead of the standard cross-entropy objective for data-imbalanced NLP tasks. It uses a dynamic weight adjustment strategy that modifies training example weights, emphasizing hard-negative examples and reducing the impact of easy-negative ones. This model achieves state-of-the-art results on the OntoNotes5.0 dataset.
- **ACE + document-context** [5] utilizes reinforcement learning-based optimization with a novel reward function to automatically find the optimal combination of embeddings for structure prediction tasks. This model achieves state-of-the-art results on CoNLL2003.

2.3.3. *Evaluation Metrics*

Like previous studies [29, 52, 55], we report the evaluation performance of each model using three standard metrics: Precision (Pre.), Recall (Rec.), and F_1, under both strict match and relaxed match conditions.

$$Pre. = \frac{TP}{TP + FP} \tag{1}$$

$$Rec. = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \frac{2 \times Pre. \times Rec.}{Pre. + Rec.} \tag{3}$$

where *TP* (true-positive) denotes the number of targets that appear in both the ground-truth and the prediction, *FP* (false-positive) denotes the number of targets that are in the prediction but not in the ground-truth, while *FN* (false-negative) denotes the number of targets that appear in the ground-truth but not appear in prediction.

Strict match refers to an exact match between the recognized entities and the ground-truth entities, while *relaxed match* [47–53] allows for some overlap between the recognized entities and the ground-truth entities.

2.3.4. *Implementation Details*

We use the GPT-3.5 (gpt-3.5-turbo-instruct) model as our LLMs backbone for all our experiments. This model supports a 4K-token context window. To maximize the utility of this capacity, we set the maximum output length to 400 tokens. Additionally, we set the temperature parameter to 0 so as to ensure reproducibility.

In addition, we use an open-source LLM, llama3 8B [56] to compare. Also, we set the temperature parameter to 0 and the maximum output length to 400 tokens.

All our experiments are conducted on a server equipped with two Intel Xeon Gold 6240R CPUs (2.40 GHz, 24 cores), 251GB of memory, and two NVIDIA RTX A5000 GPUs (24GB VRAM), running CentOS Linux 7 (Core). The server environment includes CUDA 12.1 and Python 3.7.5.

3. Results and Discussion

We evaluate the effectiveness of GPT4NER on two benchmark datasets, CoNLL2003 [45] and OntoNotes5.0 [46], against five representative state-of-the-art models. These include three few-shot models, namely ProML [38], CONTaiNER [37], and PromptNER [30], and two fully supervised models, MRC-NER + DSC [55] and ACE + document-context [5].

3.1. Experimental Results

We present experimental results on two tasks: (1) named entity recognition (NER), which aims to extract named entities from free text and then categorize them into predefined types, and (2) NEE, which is also known as entity

boundary detection that aims to simply extract named entities from free text without classifying them into specific types.

3.1.1. *Experimental Results on NER*

Table 2 presents the overall performance of GPT4NER and the five baselines on the two benchmark datasets in the NER task. For the three few-shot baselines, we include results reported in their original papers as well as results reproduced in our study, marked with*. Compared to the three few-shot baselines, among the total 12 measures (i.e., 3 metrics × 2 match types × 2 datasets), GPT4NER achieves the best performance in 10 measures and second-best in 12 measures, except for Pre. and F_1 under relaxed match on CoNLL2003. Specifically, GPT4NER achieves an F_1 score of 83.15% under

Table 2. Overall performance of GPT4NER and baselines in named entity recognition.

Dataset	Method	Strict Match			Relaxed Match		
		Pre.	*Rec.*	F_1	*Pre.*	*Rec.*	F_1
CoNLL2003	BERT_MRC + DSC	**93.41**	**93.25**	93.33	-	-	-
	ACE + document-context	-	-	**94.60**	-	-	-
	ProML(1shot)	-	-	69.16	-	-	-
	ProML(5shot)	-	-	79.16	-	-	-
	CONTaiNER(1shot)	-	-	57.80	-	-	-
	CONTaiNER(5shot)	-	-	72.80	-	-	-
	PromptNER	-	-	78.62	-	-	-
	ProML(1shot)*	63.26	65.05	64.10	76.79	78.97	77.81
	ProML(5shot)*	77.60	80.15	78.84	**85.28**	88.08	**86.65**
	CONTaiNER(1shot)*	61.47	61.10	61.27	66.80	66.42	66.59
	CONTaiNER(5shot)*	72.42	74.89	73.62	77.91	81.58	79.21
	PromptNER*	66.68	70.13	68.36	69.71	73.32	71.47
	GPT4NER-llama3	67.85	72.93	70.30	72.21	77.62	74.82
	GPT4NER(ours)	**79.20**	**87.52**	**83.15**	81.56	**90.12**	85.63
	GPT4NER w/o POS	78.24	86.05	81.96	80.96	89.04	84.81
OntoNotes5.0	BERT_MRC + DSC	**91.59**	**92.56**	**92.07**	-	-	-
	ProML(1shot)	-	-	45.98	-	-	-
	ProML(5shot)	-	-	63.24	-	-	-
	CONTaiNER(1shot)	-	-	32.00	-	-	-
	CONTaiNER(5shot)	-	-	56.20	-	-	-
	ProML(1shot)*	36.53	51.59	42.74	52.42	74.17	61.39
	ProML(5shot)*	50.21	64.46	56.42	65.91	84.80	74.13
	CONTaiNER(1shot)*	40.92	33.61	36.84	61.68	50.30	55.31
	CONTaiNER(5shot)*	54.49	53.64	54.06	73.47	72.45	72.95
	GPT4NER-llama3	37.52	55.63	44.82	45.21	67.02	53.99
	GPT4NER(ours)	62.66	71.32	66.71	74.62	84.93	79.44
	GPT4NER w/o POS	**67.15**	**73.92**	**70.37**	**79.85**	**87.90**	**83.68**

Within each type of methods, the best results are in bold and the second best are underlined. Results marked with *indicate our reproduction. Results of ProML and container on OntoNotes5.0 are reported based on the average of three splits.

strict match on CoNLL2003 and 70.37% on OntoNotes5.0, surpassing few-shot baselines by at least 4.0 points and 7.1 points on the two datasets, respectively. Under relaxed match, GPT4NER achieves the F_1 of 83.68% on OntoNotes5.0, outperforming few-shot baselines by at least 9.5 points. On CoNLL2003, GPT4NER achieves the F_1 of 85.63%, which is slightly below the best result of few-shot baselines (i.e., 86.65%). Compared to the two fully supervised baselines, GPT4NER achieves 87.9% of their best performance on CoNLL2003 and 76.4% of their best performance on OntoNotes5.0 in terms of the F_1 under strict match. Compared to llama3 8B model, GPT4NER outperforms at least 12.8 points under strict match and 10.8 points under relaxed match on CoNLL2003. On Ontonotes5.0, GPT4NER outperforms llama3 8B by 25.5 points under strict match and 29.6 points under relaxed match.

GPT4NER vs. Few-Shot Baselines. Let us compare GPT4NER to few-shot baselines. Table 2 illustrates that under strict match, GPT4NER significantly outperforms all three few-shot baselines across all three metrics on both datasets. Specifically, GPT4NER improves F_1 by at least 3.99 points on CoNLL2003 and at least 7.13 points on OntoNotes5.0. This demonstrates GPT4NER's superior ability to accurately generate named entities with predefined types. Under relaxed match, GPT4NER also outperforms all three few-shot baselines across all three metrics on both datasets, with the exception of ProML (5-shot) on CoNLL2003 in terms of Pre. and Rec. Notably, GPT4NER achieves the highest Rec. on both datasets, indicating its strong capability to generate named entities with predefined types under lenient condition.

GPT4NER vs. Fully Supervised Baselines. The two fully supervised baselines achieve state-of-the-art performance on both CoNLL2003 and OntoNotes5.0 by leveraging large amounts of annotated training data. As shown in Table 2, GPT4NER trails behind the best performance of the fully supervised baselines by 11.5 points on CoNLL2003 and by 21.7 points on OntoNotes5.0. However, fully supervised baselines [5, 55] require extensive annotated training data and perform poorly with less training data. The performance of supervised models increase with the training data [29]. By contrast, GPT4NER uses only a few labeled examples with minimal human effort in prompting LLMs, but still achieves 87.9% of fully supervised baselines' best performance on CoNLL2003 and 76.4% on OntoNotes. This demonstrates the potential of GPT4NER for few-shot NER, especially in low-resource scenarios.

Strict Match vs. Relaxed Match. Table 2 shows that all few-shot models perform better under relaxed match compared to strict match across all metrics and datasets. It shows that for all four models, the scores under relaxed match are significantly higher than the corresponding ones under strict match across all three metrics on both datasets. To illustrate the usefulness of utilizing relaxed match in addition to strict match for evaluating performance, we define a metric called

"**score improvement** (SI)" as Eq. (4) to denote the difference between the scores under relaxed match and strict match achieved by a model on a dataset:

$$\mathrm{SI(m)} = \mathrm{Relaxed(m)} - \mathrm{Strict(m)} \tag{4}$$

where Relaxed denotes the score under relaxed match, Strict denotes the score under strict match, and m ∈ {Pre.,Rec.,F_1}. For example, GPT4NER achieves SI(F_1) = 2.48 on CoNLL2003 (85.63 - 83.15).

As shown in Table 3, the four few-shot models achieve the SI(Pre.) of 2.36~13.53 points, the SI(Rec.) of 2.60~13.92 points, and the SI(F_1) of 2.48~13.71 points on CoNLL2003. On OntoNotes5.0, the SI(Pre.) values are 7.69~20.76 points, the SI(Rec.) values are 11.39~22.58 points, and the SI(F_1) values are 9.17~18.89 points. These high SI(·) values indicate that models may struggle to exactly recognize the boundaries of named entities but can partially recognize these named entities. Additionally, the SI(Pre.), SI(Rec.), and SI(F_1) values on OntoNotes5.0 are significantly higher than the corresponding ones on CoNLL2003. This difference could be due to the more complex and diverse text in OntoNotes5.0, which includes more syntactic and semantic variations. The few-shot models find it challenging to accurately recognize or generate the precise boundaries of entities in such complex and diverse texts, leading to a noticeable performance difference between relaxed match and strict match.

Table 3. SI value of GPT4NER and baselines in named entity recognition (NER).

Dataset	**Method**	**SI**		
		Pre.	**Rec.**	**F_1**
CoNLL2003	ProML(1shot)*	13.53	13.92	13.71
	ProML(5shot)*	7.68	7.93	7.81
	CONTaiNER(1shot)*	5.33	5.32	5.32
	CONTaiNER(5shot)*	5.49	6.69	5.59
	PromptNER*	3.03	3.19	3.11
	GPT4NER-llama3	4.36	4.69	4.52
	GPT4NER(ours)	**2.36**	**2.60**	**2.48**
	GPT4NER w/o POS	2.72	2.99	2.85
OntoNotes5.0	ProML(1shot)*	15.89	22.58	18.65
	ProML(5shot)*	15.70	20.34	17.71
	CONTaiNER(1shot)*	20.76	16.69	18.47
	CONTaiNER(5shot)*	18.98	18.81	18.89
	GPT4NER-llama3	**7.69**	**11.39**	**9.17**
	GPT4NER(ours)	11.96	13.61	12.73
	GPT4NER w/o POS	12.70	13.98	13.31

Within each type of methods, the smallest results are in bold and the second smallest are underlined. Results marked with *indicate our reproduction.

These high SI(Pre.), SI(Rec.), and SI(F_1) values may be attributed to the models' recognition or generation capabilities. However, annotation inconsistencies could also be a contributing factor. For example, within the same dataset, some PER/PERSON entities may include prefix words (e.g., "Mr." and "Dr."), while others may exclude these prefixes. Furthermore, some loose recognitions of entity boundaries are acceptable.

As shown in the following two examples of GPT4NER on OntoNotes5.0, the model predicts "Dick Cheney's" as a PERSON and "the Reporters' Committee for Freedom of the Press" as a ORG. The two predictions are slightly different from the corresponding ground-truth, and under strict match, they are considered wrong. However, under relaxed match, they are considered correct. This demonstrates that relaxed match evaluates performance in a broader sense and provides a more comprehensive assessment of the model, which is closer to real-world applications. Therefore, we consider relaxed match a valuable metric, complementary to strict match, for evaluating model performance.

- *Test Text*: In a separate first person account Miller confirmed that she told the grand jury that Scooter Libby Dick Cheney 's top aide discussed with her as many as three times the role of Valerie Plame as a CIA employee.
  Gold label: "Miller": "PERSON", "Scooter Libby": "PERSON", "**Dick Cheney 's**": "PERSON", "Valerie Plame": "PERSON", "CIA": "ORG"
  Prediction: "Miller": "PERSON", "Scooter Libby": "PERSON", "**Dick Cheney**": "PERSON", "Valerie Plame": "PERSON", "CIA": "ORG"

- *Test Text*: in Minneapolis Lucy Dalglish executive director of the Reporters ' Committee for Freedom of the Press.
  Gold label: "Minneapolis": "GPE", "Lucy Dalglish": "PERSON", "**the Reporters ' Committee for Freedom of the Press**": "ORG"
  Prediction: "Minneapolis": "GPE", "Lucy Dalglish": "PERSON", "**Reporters ' Committee for Freedom of the Press**": "ORG"

3.1.2. *Experimental Results on NEE*

Table 4 reports the overall performance of GPT4NER and the few-shot baselines in the NEE task. The results of the few-shot baselines are our reproductions, marked with an asterisk (*). Among the total 12 measures, GPT4NER achieves 10 best results and 9 second-best ones, except for Pre. and F_1 under relaxed match and Pre. under strict match on CoNLL2003. Specifically, GPT4NER attains an F_1 score of 88.12% under strict match on CoNLL2003 and 74.12% on OntoNotes5.0, significantly outperforming the few-shot baselines by at least 3.1 points on CoNLL2003 and at least 15.8 points on OntoNotes5.0. Under relaxed match, GPT4NER achieves an F_1 score of 90.63% on OntoNotes5.0, surpassing the few-shot baselines by at least 12.1 points. On CoNLL2003, GPT4NER achieves an F_1 score of 92.52%, which is slightly lower than the best result of the few-shot

Table 4. Performance of GPT4NER and few-shot baselines in named entity extraction (NEE) (a.k.a, entity boundary detection).

Dataset	Method	Strict Match			Relaxed Match		
		Pre.	*Rec.*	F_1	*Pre.*	*Rec.*	F_1
CoNLL2003	ProML(1shot)*	72.21	74.21	73.14	89.10	91.60	90.27
	ProML(5shot)*	83.09	85.82	84.42	92.83	95.88	**94.32**
	CONTaiNER(1shot)*	82.20	81.81	81.98	**92.97**	92.57	92.75
	CONTaiNER(5shot)*	83.54	86.45	84.96	92.38	95.60	93.95
	GPT4NER-llama3	76.66	82.38	79.42	83.97	90.24	86.99
	GPT4NER(ours)	**83.93**	**92.74**	**88.12**	88.13	**97.38**	92.52
	GPT4NER w/o POS	82.58	90.83	86.51	87.69	96.44	91.85
OntoNotes5.0	ProML(1shot)*	38.43	54.32	44.98	57.07	80.88	66.87
	ProML(5shot)*	51.84	66.59	58.27	69.78	89.85	78.50
	CONTaiNER(1shot)*	43.04	35.32	38.73	67.14	54.68	60.16
	CONTaiNER(5shot)*	56.24	55.38	55.80	77.67	76.61	77.13
	GPT4NER-llama3	41.27	61.16	49.28	53.07	78.65	63.38
	GPT4NER(ours)	66.67	75.88	70.98	82.04	93.38	87.34
	GPT4NER w/o POS	**70.72**	**77.85**	**74.12**	**86.48**	**95.20**	**90.63**

The best results are highlighted in bold and the second best are underlined. Results marked with *indicate our reproduction.

baselines (i.e., 94.32%). GPT4NER outperforms llama3 8B by 8.7 points under strict match and 5.5 points under relaxed match on CoNLL2003. On OntoNotes5.0, GPT4NER surpasses llama3 8B by 24.8 points under strict match and 27.2 points under relaxed match. These results are consistent with those reported in Section 3.1.1 and Table 2, confirming the effectiveness and robustness of GPT4NER in few-shot NER and its sub-task.

Strict Match vs. Relaxed Match. We utilize the SI as defined by Equation 4 to illustrate model performance in the NEE task. Table 5 shows that the three few-shot models achieve SI(Pre.) values of 4.20~16.89 points, SI(Rec.) values of 4.64~17.39 points, and SI(F_1) values of 4.40~17.13 points on CoNLL2003. On OntoNotes5.0, the few-shot models achieve SI(Pre.) values of 11.80~24.10 points, SI(Rec.) values of 17.35~26.56 points, and SI(F_1) values of 14.10~21.89 points. These SI(Pre.), SI(Rec.), and SI(F_1) values are consistent with those in the NER task reported in Section 3.1.1.[2] These high SI(Pre.), SI(Rec.), and SI(F_1) values confirm the usefulness and necessity of utilizing relaxed match as a complement to evaluate model performance in NER and its sub-task.

NER vs. NEE. The NER task consists of two sub-tasks: NEE and named entity classification. While previous studies primarily report the overall NER performance, we find that evaluating NEE performance separately can provide deeper insights into model capabilities. As shown in Table 4, the strict F_1 achieved by all few-shot models in the NEE sub-task is relatively low,

[2]In fact, the SI(Pre.), SI(Rec.), and SI(F_1) values in NEE are even higher than those in NER.

Table 5. SI value of GPT4NER and baselines in named entity extraction

Dataset	Method	SI		
		Pre.	*Rec.*	F_1
CoNLL2003	ProML(1shot)*	16.89	17.39	17.13
	ProML(5shot)*	9.74	10.06	9.90
	CONTaiNER(1shot)*	10.77	10.76	10.77
	CONTaiNER(5shot)*	8.84	9.15	8.99
	GPT4NER-llama3	7.31	7.86	7.57
	GPT4NER(ours)	**4.20**	**4.64**	**4.40**
	GPT4NER w/o POS	5.11	5.61	5.34
OntoNotes5.0	ProML(1shot)*	18.64	26.56	21.89
	ProML(5shot)*	17.94	23.26	20.23
	CONTaiNER(1shot)*	24.10	19.36	21.43
	CONTaiNER(5shot)*	21.43	21.23	21.33
	GPT4NER-llama3	**11.80**	17.49	**14.10**
	GPT4NER(ours)	15.37	17.50	16.36
	GPT4NER w/o POS	15.76	**17.35**	16.51

Within each type of methods, the smallest results are in bold and the second smallest are underlined. Results marked with *indicate our reproduction.

ranging from 73.14% to 88.12% on CoNLL2003 and from 38.73% to 74.12% on OntoNotes5.0. This suggests that the main factor contributing to low strict performance in NER is the low NEE performance, highlighting the need for more focus on improving NEE. Under relaxed match, all few-shot methods perform relatively well on CoNLL2003, with F_1 ranging from 90.27% to 94.32%. However, they still perform relatively poorly on OntoNotes5.0, with F_1 ranging from 60.16% to 90.63% in the NEE sub-task. This underscores the need for further improvements in NEE performance to enhance overall NER performance.

Tables 6 and 7 report detailed metrics—including precision, recall, F_1, and counts—for each entity type in both the NER and NEE tasks, comparing GPT4NER with and without the CoT module. On CoNLL2003, the chain-of-thought helps particularly on complex or ambiguous types such as ORG and MISC, where reasoning over definitions and examples may guide the model to more consistent decisions. Notably, adding the chain-of-thought often increases the number of predicted entities (Pred column) across several types. While this sometimes leads to modest gains in recall, the number of correct predictions (Correct column) does not always increase proportionally. As a result, precision can decrease and F_1 may not improve substantially. On OntoNotes5.0, the chain-of-thought similarly increases the number of predicted entities for many types (e.g., PERSON, NORP, WORK_OF_ART), but recall gains are limited and precision often drops, indicating that additional reasoning may introduce spurious entities without substantially improving coverage. This suggests that the chain-of-thought can encourage the model to identify more potential

Table 6. Detailed comparison of GPT4NER and its chain-of-thought ablation in named entity recognition, by entity type, including precision, recall, F_1 under strict match, and counts. Specifically, GPT4NER with POS tags serves as the baseline for CoNLL2003, while GPT4NER without POS tags serves as the baseline for OntoNotes5.0.

Dataset	Method	Entity Type	Strict Match			Entity Count		
			Pre.	*Rec.*	F_1	*Gold*	*Pred*	*Correct*
CoNLL2003	**GPT4NER**	PER	93.52	94.56	94.03	1617	1635	1529
		LOC	88.44	90.35	89.38	1668	1704	1507
		ORG	69.97	87.66	77.82	1661	2081	1456
		MISC	55.34	64.25	59.46	702	815	451
	w/o Chain-of-thought	PER	94.77	94.12	94.45	1617	1606	1522
		LOC	70.53	94.96	80.94	1668	2246	1584
		ORG	68.83	64.48	66.58	1661	1556	1071
		MISC	63.98	63.25	63.61	702	694	444
OntoNotes5.0	**GPT4NER**	PERSON	76.20	74.90	75.55	1988	1954	1489
		ORG	64.57	64.18	64.38	1795	1784	1152
		LOC	21.46	59.22	31.50	179	494	106
		NORP	64.79	78.12	70.84	841	1014	657
		GPE	90.82	84.33	87.45	2240	2080	1889
		FAC	28.46	27.41	27.92	135	130	37
		EVENT	20.31	41.27	27.23	63	128	26
		PRODUCT	17.93	68.42	28.42	76	290	52
		LANGUAGE	55.56	68.18	61.22	22	27	15
		WORK_OF_ART	36.44	75.30	49.12	166	343	125
	w/o Chain-of-thought	PERSON	80.04	77.87	78.94	1988	1934	1548
		ORG	65.60	65.13	65.36	1795	1782	1170
		LOC	25.60	59.22	35.75	179	414	106
		NORP	75.34	78.83	77.05	841	880	663
		GPE	91.21	84.82	87.90	2240	2083	1900
		FAC	26.88	37.04	31.15	135	186	50
		EVENT	19.61	31.75	24.24	63	102	20
		PRODUCT	29.71	68.42	41.43	76	175	52
		LANGUAGE	62.96	77.27	69.39	22	27	17
		WORK_OF_ART	47.79	71.69	57.35	166	249	119

entities, but the overall benefit depends on dataset complexity, entity distribution, and context length.

3.1.3. Ablation Study

In our ablation study, we analyze the impact of each component in GPT4NER by systematically removing them one at a time and observing the model performance on both NER and NEE tasks. The best-performing configurations serve as baselines for these experiments. Specifically, GPT4NER with POS tags serves as the baseline for CoNLL2003, while GPT4NER without POS tags serves as the baseline for OntoNotes5.0. The results of these ablation experiments for the NER task are presented in Table 8, and the results for the NEE task are reported in Table 9.

Table 7. Detailed comparison of GPT4NER and its chain-of-thought ablation in named entity extraction, by entity type, including precision, recall, F_1 under strict match, and counts. Specifically, GPT4NER with POS tags serves as the baseline for CoNLL2003, while GPT4NER without POS tags serves as the baseline for OntoNotes5.0.

Dataset	Method	Entity Type	Strict Match			Entity Count		
			Pre.	*Rec.*	F_1	*Gold*	*Pred*	*Correct*
CoNLL2003	**GPT4NER**	PER	94.74	95.79	95.26	1617	1635	1549
		LOC	94.95	97.00	95.97	1668	1704	1618
		ORG	72.32	90.61	80.44	1661	2081	1505
		MISC	69.45	80.63	74.62	702	815	566
	w/o Chain-of-thought	PER	96.45	95.79	96.12	1617	1606	1549
		LOC	71.86	96.76	82.47	1668	2246	1614
		ORG	99.16	92.90	95.93	1661	1556	1543
		MISC	79.97	79.06	79.51	702	694	555
OntoNotes5.0	**GPT4NER**	PERSON	77.84	76.51	77.17	1988	1954	1521
		ORG	69.67	69.25	69.46	1795	1784	1243
		LOC	22.87	63.13	33.58	179	494	113
		NORP	66.77	80.50	72.99	841	1014	677
		GPE	95.14	88.35	91.62	2240	2080	1979
		FAC	66.92	64.44	65.66	135	130	87
		EVENT	20.31	41.27	27.23	63	128	26
		PRODUCT	17.93	68.42	28.42	76	290	52
		LANGUAGE	62.96	77.27	69.39	22	27	17
		WORK_OF_ART	37.32	77.11	50.29	166	343	128
	w/o Chain-of-thought	PERSON	81.44	79.23	80.32	1988	1934	1575
		ORG	69.68	69.14	69.41	1795	1781	1241
		LOC	28.02	64.80	39.12	179	414	116
		NORP	77.73	81.33	79.49	841	880	684
		GPE	96.54	89.78	93.04	2240	2083	2011
		FAC	46.24	63.70	53.58	135	186	86
		EVENT	25.49	41.27	31.52	63	102	26
		PRODUCT	29.71	68.42	41.43	76	175	52
		LANGUAGE	70.37	86.36	77.55	22	27	19
		WORK_OF_ART	51.00	76.51	61.20	166	249	127

Table 8. Ablation study in the named entity recognition task. The best results are in bold.

Dataset	Method	Strict Match			Relaxed Match		
		Pre.	*Rec.*	F_1	*Pre.*	*Rec.*	F_1
CoNLL2003	**GPT4NER**	**79.20**	**87.52**	**83.15**	**81.56**	**90.12**	**85.63**
	w/o Entity definition	77.64	86.99	82.05	80.50	90.19	85.07
	w/o Few-shot examples	23.12	46.09	30.79	32.83	65.46	43.73
	w/o Chain-of-thought	75.57	81.82	78.57	77.66	84.08	80.74
OntoNotes5.0	**GPT4NER**	67.15	73.92	70.37	79.85	87.90	83.68
	w/o Entity definition	63.48	72.35	67.63	76.07	86.70	81.04
	w/o Few-shot examples	36.76	45.80	40.78	44.33	55.23	49.18
	w/o Chain-of-thought	**71.87**	**75.22**	**73.50**	**84.28**	**88.21**	**86.20**

Table 9. Ablation study on the named entity extraction task. The best results are in bold.

Dataset	Method	Strict Match			Relaxed Match		
		Pre.	*Rec.*	F_1	*Pre.*	*Rec.*	F_1
CoNLL2003	**GPT4NER**	83.93	92.74	88.12	88.13	**97.38**	92.52
	w/o Entity definition	82.08	91.96	86.74	86.66	97.10	91.58
	w/o Few-shot examples	31.90	63.60	42.49	46.67	93.04	62.16
	w/o Chain-of-thought	**86.03**	**93.15**	**89.45**	**89.84**	97.27	**93.41**
OntoNotes5.0	**GPT4NER**	70.72	77.85	74.12	86.48	**95.20**	90.63
	w/o Entity definition	66.54	75.84	70.89	83.36	95.02	88.81
	w/o Few-shot examples	41.17	51.29	45.67	51.49	64.14	57.13
	w/o Chain-of-thought	**75.58**	**79.11**	**77.30**	**90.64**	94.87	**92.71**

Impact of Few-Shot Examples. Table 8 illustrates the significant impact of few-shot examples on the performance of GPT4NER in the NER task. When these examples are removed, the F_1 score of GPT4NER drop substantially by 52.38 points under strict match and 41.90 points under relaxed match on CoNLL2003, and by 29.59 points under strict match and 34.50 points under relaxed match on OntoNotes5.0. Similarly, Table 9 shows that the F_1 score in NEE decreases by 45.63 points under strict match and 30.36 points under relaxed match on CoNLL2003, and by 28.45 points under strict match and 33.50 points under relaxed match on OntoNotes5.0. These notable decreases in F_1 scores across both matches, tasks, and datasets underscore the critical role of few-shot examples in the performance of GPT4NER.

Conversely, without few-shot examples, GPT4NER essentially operates as a zero-shot model. The results indicate that the introduction of just few-shot examples with minimal human effort can lead to substantial performance improvements in both NER and NEE tasks.

Furthermore, despite the explicit specification of the output format in this experiment, the absence of the implicit output format in the examples led to some generated results having correct content but incorrect format. This inconsistency affects the reliability of subsequent evaluation, as illustrated below:

- *Test Text*: This is **Xu Li**.
  Gold label: "**Xu Li**": "**PERSON**"
  Prediction: 1. **Xu Li | True | Xu Li is a proper name, making it a PERSON entity.**

In this example, "Xu Li" is correctly identified and classified, but the output does not follow format requirements, and the labeling fails in the subsequent processing.

Impact of Entity Definitions. Table 8 reveals that removing entity definition from GPT4NER results in a slight decline in F_1 performance for the NER task: a decrease of 1.10 points under strict match and 0.56 points under

relaxed match on CoNLL2003, and a decrease of 2.74 points under strict match and 2.64 points under relaxed match on OntoNotes5.0. Similarly, Table 9 shows that the F_1 scores for NEE decrease by 1.38 points under strict match and 0.94 points under relaxed match on CoNLL2003, and by 3.23 points under strict match and 1.82 points under relaxed match on OntoNotes5.0. These decreases indicate that entity definition is beneficial for both NER and NEE tasks in GPT4NER. However, these F_1 decreases are relatively minor compared to the significant drops caused by removing few-shot examples. A possible reason is that LLMs like GPT-3.5 can infer full or partial entity definitions from the input few-shot examples. This suggests that LLMs possess the capability to deduce abstract concepts from specific instances.

Impact of Chain-of-Thought. Table 8 shows that removing the CoT component from GPT4NER leads to a decrease in F_1 performance in NER by 4.58 points under strict match and 4.89 points under relaxed match on CoNLL2003. Conversely, without the CoT component, GPT4NER's performance increases by 3.13 points under strict match and 2.52 points under relaxed match on OntoNotes5.0. Table 9 further indicates that without the CoT component, GPT4NER achieves consistent increases in F_1 performance for the NEE task by 1.33 points under strict match and 0.89 points under relaxed match on CoNLL2003, and by 3.18 points under strict match and 2.08 points under relaxed match on OntoNotes5.0. These mixed results suggest that CoT prompting can be both beneficial and detrimental for few-shot models in NER and NEE tasks. A possible reason for this inconsistency is that CoT prompting might inadvertently accumulate errors. This implies that while CoT prompting has potential, its effective design remains challenging and is not always advantageous. The CoT module is originally designed to improve the interpretability of the model, but this module requires the model to add a reason for determining the entity type in the output, which really increases the output complexity of the model.

Impact of POS Tags. Table 2 shows that removing POS tags from GPT4NER leads to a decrease in F_1 performance in NER by 1.1 points under strict match and 0.8 points under relaxed match on CoNLL2003. However, it results in an increase of 3.6 points under strict match and 4.2 points under relaxed match on OntoNotes5.0. Similarly, Table 4 reveals that in the NEE task, the performance of GPT4NER without POS tags decreases by 1.6 points under strict match and 0.6 points under relaxed match on CoNLL2003, but increases by 3.1 points under strict match and 3.2 points under relaxed match on OntoNotes5.0. These results indicate that POS tags are consistently beneficial for CoNLL2003 across both NER and NEE tasks and both match metrics. Conversely, they are consistently detrimental for OntoNotes5.0 across both tasks and match metrics. A possible explanation for this discrepancy is that in datasets like CoNLL2003, where entity boundaries are

Table 10. Top-5 POS tag distributions in CoNLL2003 and OntoNotes5.0.

Dataset	POS Tag	Percent.
CoNLL2003	NN	18.69%
	FW	18.04%
	NNP	10.26%
	UH	9.88%
	CD	7.14%
OntoNotes5.0	NN	22.10%
	FW	19.56%
	UH	11.20%
	NNP	7.40%
	GW	6.24%

CD: Cardinal Number, FW: Foreign Word, GW: Goes With, NN: Common Noun, NNP: Proper Noun, UH: Interjection

clearer and sentence structures more regular, POS tags provide valuable contextual information. By contrast, in datasets like OntoNotes5.0, where entity boundaries are more ambiguous and sentence structures are more diverse and complex, POS tags may introduce noise that negatively affects model performance. Table 10 provides additional insight into the linguistic differences between the datasets. CoNLL2003 is dominated by tags such as NN, NNP, and CD, which offer clear cues for entity recognition. OntoNotes5.0, by contrast, has a higher proportion of NN and FW, with a wider variety of entity types and more complex sentence structures, reducing the effectiveness of POS information and occasionally introducing noise. These observations suggest that POS tags can be beneficial for datasets with clearer entity boundaries and regular sentence structures, such as CoNLL2003, but may not generalize to datasets with more ambiguous boundaries, complex syntax, or fine-grained labels, such as OntoNotes5.0. Careful consideration of dataset characteristics is thus recommended when incorporating POS features in few-shot NER and NEE tasks.

3.2. Error Analysis

There are three main types of errors in the evaluation of GPT4NER:

(1) **Post-Processing Errors**. The addition of POS tags often leads to the omission of spaces around hyphens and possessive markers on OntoNotes5.0, making it difficult to locate the corresponding phrases in test text for annotation, as illustrated below:

- *Test Text*: Does the President still believe that **Kim Jong- Il** is a tyrant a pygmy and a spoiled child.
  Gold label: "**Kim Jong- Il**": "PERSON"
  Prediction: **Kim Jong-Il** | True | as it is a person's name (PERSON)

- *Test Text*: in Minneapolis Lucy Dalglish executive director of **the Reporters ' Committee for Freedom of the Press** .
  Gold label: "Minneapolis": "GPE", "Lucy Dalglish": "PERSON", "**the Reporters ' Committee for Freedom of the Press**": "ORG"
  Prediction: **Reporters' Committee for Freedom of the Press** | True | as it is the name of an organization (ORG)

When processing test texts with POS tags, LLMs tend to focus on the POS of connectors. When generating output, they often omit spaces around connectors to follow formal expressions. However, this can introduce issues for subsequent processing.

(2) **Hallucination Errors**. GPT4NER occasionally returns entity types that are not included in the entity-definition component. This issue is particularly evident in ablation experiments when few-shot examples are removed, which increases the likelihood of hallucinations in LLMs, as illustrated below:

- *Test Text*: At present, we should not have a problem with watching television .
  Gold label: None
  Prediction: "present": "TIME", "problem": "PROBLEM", "television": "PRODUCT"
- *Test Text*: And let me go back to January of two thousand two in the President 's axis of evil speech before congress .
  Gold label: "congress": "ORG"
  Prediction: "January": "DATE", "two thousand two": "DATE", "President": "TITLE", "axis of evil": "PHRASE", "congress": "ORG"

Hallucinations often occur in experiments where POS tags are added or few-shot examples are removed. POS tags introduce additional complexity, and without few-shot examples, the model's learning becomes less robust.

(3) **Annotation Errors**. These errors often stem from annotator oversight. Despite rigorous review and proofreading, minor mistakes are inevitable, as illustrated below:

- *Test Text*: JAPAN GET LUCKY WIN, **CHINA** IN SURPRISE DEFEAT
  Gold label: "JAPAN": "LOC", "**CHINA**": "PER"
- *Test Text*: Rumsfeld: The Iraqis received us with overwhelming happiness and welcomed us, because of the practices of your bloody regime over the course of all those years during which you governed **Iraq**.
  Gold label: "Rumsfeld": "PERSON", "Iraqis": "NORP", "**Iraq**": "PERSON"

Obviously, “CHINA” and “Iraq” refer to countries or locations instead of persons.

3.3. Limitations

There are two primary limitations in our work. The first concerns interpretability. Depending solely on ChatGPT’s reasoning within candidate entities may not provide sufficient interpretability. LLMs such as ChatGPT introduce uncertainty in their generated output, and explanations provided in the results may not always be accurate or reliable. The second limitation pertains to token constraints. While using larger models like GPT-4 and GPT-5 may enhance performance, this is not our main focus.

4. Conclusion

This article introduces GPT4NER, a method based on LLMs for few-shot NER. GPT4NER constructs effective prompts using entity definition, few-shot examples, CoT, and POS tags to leverage the capabilities of LLMs in transforming the few-shot NER task into a sequence-generation task. Experimental results on two benchmark datasets demonstrate that GPT4NER significantly outperforms state-of-the-art few-shot models and achieves competitive results compared to fully supervised models. Furthermore, our experiments advocate for the use of the relaxed-match metric (which is widely used in time expression recognition and normalization) to evaluate model performance. Additionally, our experiments also suggest to report performance in the NEE sub-task to deepen insights into model capabilities in the NER task.

Author Contributions

Yufei Zhao: Data curation, Methodology, Investigation, Visualization, Writing—original draft; **Xiaoshi Zhong:** Conceptualization, Supervision, Writing—review & editing; **Erik Cambria:** Supervision, Writing—review & editing; **Jagath C. Rajapakse:** Supervision, Writing—review & editing.

Funding

This research was mainly supported by a startup funding (Project No.: XSQD-202007008) from Beijing Institute of Technology, China.

Ethical statement

Not applicable.

Data availability statement

Source codes and data are available at https://github.com/xszhong/GPT4NER.

Conflict of Interest

The authors declare no conflict of interest.

References

1 Chinchor N, Robinson P. Muc-7 named entity task definition. In: *Proceedings of the 7th Conference on Message Understanding* April 29 - May 1, 1998 Fairfax, Virginia, USA; 1997 [Accessed 2024 May 1] **29**; p. 1–21.

2 Devlin J, Chang M-W, Lee K, Toutanova K. 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) June 2 to June 7, 2019 (Stroudsburg, Pennsylvania, USA: Association for Computational Linguistics) Minneapolis, Minnesota, USA; 2019 [Accessed 2024 May 1]; p. 4171–4186. doi:10.18653/v1/N19-1423.

3 Yang Y, Katiyar A. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* November 16–20, 2020 Online; 2020 [Accessed 2024 May 1]; p. 6365–6375. doi:10.18653/v1/2020.emnlp-main.516.

4 Ding B, Liu L, Bing L, Kruengkrai C, Nguyen TH, Joty S, et al. Daga: data augmentation with a generation approach for low-resource tagging tasks. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP) November 16–20, 2020 Online; 2020 [Accessed 2024 May 1]; p. 6045–6057. doi:10.18653/v1/2020.emnlp-main.488.

5 Wang X, Jiang Y, Bach N, Wang T, Huang Z, Huang F, et al. Automated concatenation of embeddings for structured prediction. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume 1, Long Papers) August 1-6, 2021 Online; 2020 [Accessed 2024 May 1]; p. 2643–2660. doi:10.18653/v1/2021.acl-long.206.

6 Li J, Fei H, Liu J, Wu S, Zhang M, Teng C, et al. Unified named entity recognition as word-word relation classification. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 22 February - 1 March, 2022 Online; 2022 [Accessed 2024 May 1] **36**; p. 10965–10973. doi:10.1609/aaai.v36i10.21344.

7 Hanisch D, Fundel K, Mevissen H-T, Zimmer R, Fluck J. ProMiner: rule-based protein and gene entity recognition. *BMC Bioinformatics*. 2005;**6**(S1):1–9. doi:10.1186/1471-2105-6-S1-S14.

8 Riaz K. Rule-based named entity recognition in Urdu. In: *Proceedings of the 2010 Named Entities Workshop* 16 July 2010 Uppsala, Sweden; 2010 [Accessed 2024 May 1]; p. 126–135. doi:10.5555/1870457.1870476.

9 Sasaki Y, Tsuruoka Y, McNaught J, Ananiadou S. How to make the most of NE dictionaries in statistical NER. *BMC Bioinformatics*. 2008;**9**(S11):1–9. doi:10.1186/1471-2105-9-S11-S5.

10 Egorov S, Yuryev A, Daraselia N. A Simple and practical dictionary-based approach for identification of proteins in medline abstracts. *J Am Med Inform Assoc*. 2004;**11**(3):174–178. doi:10.1197/jamia.M1453.

11 Morwal S, Jahan N, Chopra D. Named entity recognition using hidden Markov model (HMM). *International Journal on Natural Language Computing (IJNLC)*. 2012;**1**(4):15–23.

12 Zhao S. Named Entity recognition in biomedical texts using an HMM model. In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (NLPBA/BioNLP)* August 28–29, 2004 Geneva Switzerland; 2004 [Accessed 2024 May 1]; p. 84–87. doi:10.5555/1567594.1567613.

13 Xu Z, Qian X, Zhang Y, Zhou Y. CRF-based hybrid model for word segmentation, NER and even POS Tagging. In: *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing* January 11–12, 2008 167-170. Hyderabad, India; 2008 [Accessed 2024 May 1].

14 Li L, Zhou R, Huang D. Two-phase biomedical named entity recognition using CRFs. *Comput Biol Chem*. 2009;**33**(4):334–338. doi:10.1016/j.compbiolchem.2009.07.004.

15 Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (Almost) from scratch. *J Mach Learn Res*. 2011;**12**:2493–2537. doi:10.5555/1953048.2078186.

16 Chiu JP, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. *Trans Assoc Comput Linguist*. 2016;**4**:357–370. doi:10.1162/tacl_a_00104.

17 Ma X, Hovy E. 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); 2016 [Accessed 2024 May 1]; p. 1064–1074. doi:10.18653/v1/P16-1101.

18 Lyu C, Chen B, Ren Y, Ji D. Long short-term memory RNN for biomedical named entity recognition. *BMC Bioinformatics*. 2017;**18**(1):1–11. doi:10.1186/s12859-017-1868-5.

19 Chowdhury S, Dong X, Qian L, Li X, Guan Y, Yang J, et al. A multitask bi-directional RNN model for named entity recognition on Chinese electronic medical records. *BMC Bioinformatics*. 2018;**19** (S17):75–84. doi:10.1186/s12859-018-2467-9.

20 Wang C, Liu X, Chen Z, Hong H, Tang J, Song D. DeepStruct: pretraining of language models for structure prediction. In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022. pp. 803–823. doi:10.18653/v1/2022.findings-acl.67.

21 Chen J, Lu Y, Lin H, Lou J, Jia W, Dai D, Wu H, Cao B, Han X, Sun L. Learning in-context learning for named entity recognition. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* Volume 1, Long Papers; 2023 [Accessed 2024 May 1]; p. 13661–13675. doi:10.18653/v1/2023.acl-long.764.

22 Hu X, Jiang Y, Liu A, Huang Z, Xie P, Huang F, et al. Entity-to-text based data augmentation for various named entity recognition tasks. *Findings Assoc Comput Linguist ACL*. 2023;9072–9087. doi:10.18653/v1/2023.findings-acl.578.

23 Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. *Adv Neural Inf Process Syst*. 2020:1877–1901. doi:10.5555/3495724.3495883.

24 Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, et al. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774. 2023. doi:10.48550/arXiv.2303.08774.

25 Wadhwa S, Amir S, Wallace BC. Revisiting relation extraction in the era of large language models. In: *Proceedings of the Conference. Association for Computational Linguistics*. Meeting July 9-14, 2023 Toronto, Canada. NIH Public Access; 2023 [Accessed 2024 May 1] **2023**; p. 15566-15589. doi:10.18653/v1/2023.acl-long.868.

26 Dagdelen J, Dagdelen J, Lee S, Lee S, Rosen AS, Ceder G, et al. Structured information extraction from complex scientific text with fine-tuned large language models[J]. *Nat Commun*. 2024; **15**(1):1418. doi:10.1038/s41467-024-45563-x.

27 Lu P, Mishra S, Xia T, Qiu L, Chang K-W, Zhu S-C, et al. Learn to explain: multimodal reasoning via thought chains for science question answering. *Adv Neural Inf Process Syst*. 2022;**35**:2507–2521. doi:10.5555/3600270.3600452.

28 Huang Y, He K, Wang Y, Zhang X, Gong T, Mao R, et al. COPNER: contrastive learning with prompt guiding for few-shot named entity recognition. In: *Proceedings of the 29th International Conference on Computational Linguistics* October 12-17, 2022 Gyeongju, Republic of Korea; 2022 [Accessed 2024 May 1]; p. 2515–2527.

29 Wang S, Sun X, Li X, Ouyang R, Wu F, Zhang T, et al.: GPT-NER: Named Entity Recognition via Large Language Models. arXiv preprint arXiv:2304.10428. 2023. doi:10.48550/arXiv.2304.10428.

30 Ashok D, Lipton ZC. PromptNER: Prompting for Named Entity Recognition. arXiv preprint arXiv:2305.15444. 2023. doi:10.48550/arXiv.2305.15444.

31 Xie T, Li Q, Zhang J, Zhang Y, Liu Z, Wang H. Empirical study of zero-shot NER with ChatGPT. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* December 6–10, 2023 Singapore; 2023 [Accessed 2024 May 1]; p. 7935–7956. doi:10.18653/v1/2023.emnlp-main.493.

32 Hu Y, Chen Q, Du J, Peng X, Kuttichi Keloth V, Zuo X, et al. Improving large language models for clinical named entity recognition via prompt engineering. *J Am Med Inform Assoc*. 2024;**31**(9): 1812–1820. doi:10.1093/jamia/ocad259.

33 Shao W, Zhang R, Ji P, Fan, D, Hu, Y, Yan, X, et al. Astronomical knowledge entity extraction in astrophysics Journal articles via large language models. *Res Astron Astrophys*. 2024;**24**(6):065012. doi:10.1088/1674-4527/ad3d15.

34 Ghosh S, Tyagi U, Kumar S, Manocha D. BioAug: conditional generation based data augmentation for low-resource biomedical NER. In: *SIGIR '23: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* July 23 - 27, 2023 Taipei Taiwan; 2023 [Accessed 2024 May 1]; p. 1853–1858. doi:10.1145/3539618.3591957.

35 Ye J, Xu N, Wang Y, Zhou J, Zhang Q, Gui T, et al. LLM-DA: data augmentation via large language models for few-shot named entity recognition. arXiv preprint arXiv:2402.14568. 2024. doi:10.48550/arXiv.2402.14568.

36 Zhao D, Mu W, Jia X, Liu, S, Chu, Y, Meng, J, et al. Few-shot biomedical NER empowered by LLMs-assisted data augmentation and multi-scale feature extraction. *BioData Mining*. 2025;**18**(1):28. doi:10.1186/s13040-025-00443-y.

37 Das SSS, Katiyar A, Passonneau R, Zhang R. CONTaiNER: few-shot named entity recognition via contrastive learning. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (Volume 1, Long Papers) May 22 – 27, 2022 Dublin, Ireland; 2022 [Accessed 2024 May 1]; p. 6338–6353. doi:10.48550/arXiv.2109.07589.

38 Chen Y, Zheng Y, Yang Z. Prompt-based metric learning for few-shot NER. *Findings Assoc Comput Linguist ACL*. 2023;7199–7212. doi:10.18653/v1/2023.findings-acl.451.

39 Zhou W, Zhang S, Gu Y, Chen M, Poon H: UniversalNER: targeted distillation from large language models for open named entity recognition. arXiv preprint arXiv:2308.03279. 2023. doi:10.48550/arXiv.2308.03279.

40 Layegh A, Payberah AH, Soylu A, Roman D, Matskin M. ContrastNER: contrastive-based prompt tuning for few-shot NER. In: *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)* June 26 - 30, 2023 Torino, Italy. Piscataway, New Jersey, USA: IEEE; 2023 [Accessed 2024 May 1]; p. 241–249. doi:10.1109/COMPSAC57700.2023.00038.

41 Wei J, Wang X, Schuurmans D, Bosma, M, Ichter, B, Xia, F, et al 28 November. Chain-of-thought prompting elicits reasoning in large language models. In: *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems* November 28 to December 9, 2022 New Orleans LA USA; 2022 [Accessed 2024 May 1] **35**; p. 24824–24837. doi:10.5555/3600270.3602070.

42 Zhou D, Schärli N, Hou L, Wei J, Scales N, Wang X, et al. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. arXiv preprint arXiv:2205.10625. 2022. doi:10.48550/arXiv.2205.10625.

43 Zhang Z, Zhang A, Li M, Smola A: Automatic Chain of Thought Prompting in Large Language Models. arXiv preprint arXiv:2210.03493. 2022. doi:10.48550/arXiv.2210.03493.

44 Guo Q, Dong Y, Tian L, et al. BANER: boundary-aware LLMs for few-shot named entity recognition. In: *Proceedings of the 31st International Conference on Computational Linguistics* January 19 – 24, 2025 Abu Dhabi, UAE; 2025 [Accessed 2025 Aug 1]; p. 10375–10389. https://aclanthology.org/2025.coling-main.691.

45 Sang EF, De Meulder F: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. arXiv preprint cs/0306050. 2003. doi: 10.48550/arXiv.cs/0306050.

46 Pradhan S, Moschitti A, Xue N, Ng HT, Björkelund A, Uryupina O. Towards robust linguistic analysis using OntoNotes. In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* August 8-9, 2013 Sofia, Bulgaria; 2013 [Accessed 2025 Aug 1]; p. 143–152. https://aclanthology.org/W13-3516/.

47 Verhagen M, Gaizauskas R, Schilder F, Hepple M, Katz G, Pustejovsky J. SemEval-2007 Task 15: tempEval temporal relation identification. In: *Proceedings of the 4th International Workshop on Semantic*

Evaluation June 23–24, 2007 Prague, Czech Republic; 2007 [Accessed 2024 May 1]; p. 75–80. https://aclanthology.org/S07-1014/.

48 Verhagen M, Sauri R, Caselli T, Pustejovsky J. SemEval-2010 Task 13: tempEval-2. In: *Proceedings of the 5th International Workshop on Semantic Evaluation* July 15-16, 2010 Uppsala, Sweden; *2010* [Accessed 2024 May 1]; p. 57–62. https://aclanthology.org/S10-1010/.

49 UzZaman N, Llorens H, Derczynski L, Verhagen M, Allen J, Pustejovsky J. SemEval-2013 Task 1: tempEval-3: evaluating time expressions, events, and temporal relations. In: *Proceedings of the 7th International Workshop on Semantic Evaluation* June, 2013 Atlanta, Georgia, USA; 2013 [Accessed 2024 May 1]; p. 1–9. https://aclanthology.org/S13-2001/.

50 Zhong X, Sun A, Cambria E. Time expression analysis and recognition using syntactic token types and general heuristic rules. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. **1**. Vancouver, Canada; 2017 [Accessed 2024 May 1]; p. 420–429. doi:10.18653/v1/P17-1039.

51 Zhong X, Cambria E. Time expression recognition using a constituent-based tagging scheme. In: *Proceedings of the 2018 World Wide Web Conference*, *Lyon*, France; 2018 [Accessed 2024 May 1]; p. 983–992. doi:10.1145/3178876.3185997.

52 Zhong X, Cambria E, Hussain A. Extracting time expressions and named entities with constituent-based tagging schemes. *Cogn Comput*. 2020;**12**(4):844–862. doi:10.1007/s12559-020-09714-8.

53 Zhong X, Cambria E. Time Expression and Named Entity Recognition. In: *Socio-Affective Computing*. Amir, Hussain, Erik, Cambria Socio-Affective Computing. Singapore: Springer; 2021 [Accessed 2024 May 1] **10**; p. 96. doi:10.1007/978-3-030-78961-9.

54 Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, et al. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv preprint arXiv:2203.11171. 2022. doi:10.48550/arXiv.2203.11171.

55 Li X, Sun X, Meng Y, Liang J, Wu F, Li J. Dice loss for data-imbalanced NLP tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* July 5–10, 2020 Online; 2020 [Accessed 2024 May 1]; p. 465–476. doi:10.18653/v1/2020.acl-main.45.

56 Grattafiori A, Dubey A, Jauhri A, et al. The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783. 2024. doi:10.48550/arXiv.2407.21783.