UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

A Cooperative Multi-Agent Framework for Zero-Shot Named Entity Recognition

Wang, Zihan; Zhao, Ziqi; Lyu, Yougang; Chen, Zhumin; de Rijke, Maarten; Ren, Zhaochun

DOI
10.1145/3696410.3714923

Publication date
2025

Document Version
Final published version

Published in
WWW '25 : Proceedings of the ACM Web Conference 2025

License
CC BY

Link to publication

Citation for published version (APA):
Wang, Z., Zhao, Z., Lyu, Y., Chen, Z., de Rijke, M., & Ren, Z. (2025). A Cooperative Multi-Agent Framework for Zero-Shot Named Entity Recognition. In *WWW '25 : Proceedings of the ACM Web Conference 2025: April 28-May 2, 2025, Sydney, NSW, Australia* (pp. 4183-4195). Association for Computing Machinery. https://doi.org/10.1145/3696410.3714923

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)
Download date:12 Oct 2026

A Cooperative Multi-Agent Framework for Zero-Shot Named Entity Recognition

Zihan Wang*
University of Amsterdam
Amsterdam, The Netherlands
zhw.cypher@gmail.com

Ziqi Zhao*
Shandong University
Qingdao, China
ziqizhao.work@gmail.com

Yougang Lyu
University of Amsterdam
Amsterdam, The Netherlands
youganglyu@gmail.com

Zhumin Chen
Shandong University
Jinan, China
chenzhumin@sdu.edu.cn

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

Zhaochun Ren
Leiden University
Leiden, The Netherlands
z.ren@liacs.leidenuniv.nl

Abstract

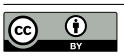
Zero-shot named entity recognition (NER) aims to develop entity recognition systems from unannotated text corpora. This task presents substantial challenges due to minimal human intervention. Recent work has adapted large language models (LLMs) for zero-shot NER by crafting specialized prompt templates. And it advances models' self-learning abilities by incorporating self-annotated demonstrations. Two important challenges persist: (i) Correlations between contexts surrounding entities are overlooked, leading to wrong type predictions or entity omissions. (ii) The indiscriminate use of task demonstrations, retrieved through shallow similarity-based strategies, severely misleads LLMs during inference.

In this paper, we introduce the *cooperative multi-agent system* (CMAS), a novel framework for zero-shot NER that uses the collective intelligence of multiple agents to address the challenges outlined above. CMAS has four main agents: (i) a self-annotator, (ii) a type-related feature (TRF) extractor, (iii) a demonstration discriminator, and (iv) an overall predictor. To explicitly capture correlations between contexts surrounding entities, CMAS reformulates NER into two subtasks: recognizing named entities and identifying entity type-related features within the target sentence. To enable controllable utilization of demonstrations, a demonstration discriminator is established to incorporate the self-reflection mechanism, automatically evaluating helpfulness scores for the target sentence. Experimental results show that CMAS significantly improves zero-shot NER performance across six benchmarks, including both domain-specific and general-domain scenarios. Furthermore, CMAS demonstrates its effectiveness in few-shot settings and with various LLM backbones.[1]

CCS Concepts

• **Computing methodologies** → **Information extraction**; **Multi-agent systems**.

*Both authors contributed equally to the paper.
[1]Our code is available at https://github.com/WZH-NLP/WWW25-CMAS.



This work is licensed under a Creative Commons Attribution 4.0 International License.
WWW '25, April 28-May 2, 2025, Sydney, NSW, Australia
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714923

Keywords

Information extraction, Named entity recognition, Zero-shot learning, Large language models, Multi-agent systems

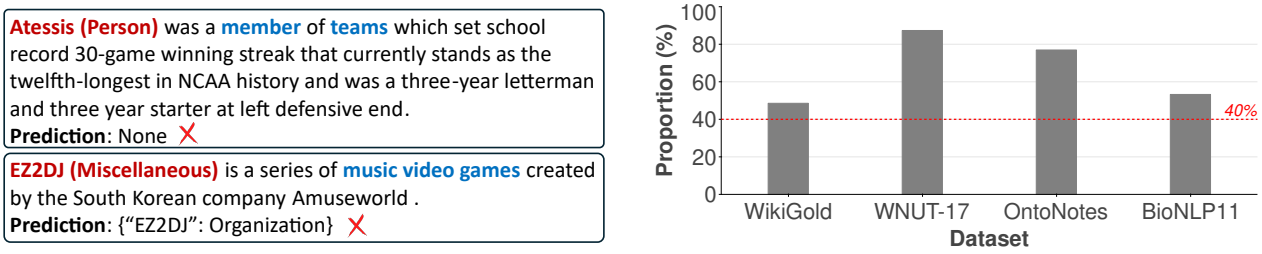
ACM Reference Format:
Zihan Wang, Ziqi Zhao, Yougang Lyu, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2025. A Cooperative Multi-Agent Framework for Zero-Shot Named Entity Recognition. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3696410.3714923

1 Introduction

In named entity recognition (NER), the task is to identify predefined named entities, such as persons, locations, and organizations, based on their contextual semantics within input texts. NER serves as a fundamental task in information extraction and is integral to various downstream natural language processing (NLP) applications, including question answering [17, 37], document understanding [12], and information retrieval [5, 6]. Current NER methods primarily use fully supervised learning paradigms and show impressive performance across various benchmarks [30, 39, 51]. However, these fully-supervised paradigms rely heavily on large-scale, human-annotated data. In real-world scenarios, the availability of annotated data may be restricted to specific domains, severely hindering the generalizability and adaptability of NER models [47, 60].

Recently, large language models (LLMs) have transformed natural language processing with their zero-shot or few-shot generalization abilities [3, 24, 26, 27]. With their extensive search spaces and large-scale pre-training data, LLMs have the potential to overcome the challenges of data sparsity and generalizability faced by NER models. Motivated by these developments, one line of prior work explores prompting techniques to enhance few-shot in-context learning (ICL) for NER [45]. Other efforts use specialized knowledge to develop task-specific LLMs for NER [35, 60, 62] or employ LLMs as data annotators or generators to augment smaller language models [14, 22]. However, these approaches still require deliberately selected annotated task examples or external knowledge. The reasoning abilities of LLMs for zero-shot NER remain underexplored.

To address zero-shot NER, Wei et al. [49] transform this task, where no labeled data is available, into a two-stage question-answering process by chatting with LLMs. Xie et al. [54] conduct a systematic empirical study on zero-shot NER using LLMs and tailor



(a) Overlooking correlated contexts surrounding entities.

(b) Proportions of demonstrations without target entity types.

Figure 1: (a) Examples of incorrect type prediction and entity omission from an existing method [54] due to overlooking correlated contexts surrounding entities. Red texts represent wrongly recognized entities; golden labels are included in the brackets. Blue texts highlight entity type-related features (TRFs), i.e., contexts strongly associated with the entity types. (b) Proportions of selected demonstrations lacking target entity types in the WikiGold [2], WNUT-17 [4], OntoNotes,[2] and BioNLP11 [31] datasets. More than 40% of demonstrations do not contain any entity types within the target sentence.

prevalent reasoning methods, such as tool augmentation and majority voting, to adapt ChatGPT for zero-shot NER. Furthermore, to reduce reliance on external tools, Xie et al. [55] enhance the self-learning capabilities of LLMs through a self-improvement mechanism. Specifically, LLMs initially annotate unlabeled corpora with self-consistency scores. Subsequently, for each test input, inference is conducted using in-context learning (ICL) with demonstrations retrieved from the self-annotated dataset.

Despite these advances, current zero-shot NER methods still encounter two challenging problems:

Challenge 1: Overlooking correlations between contexts surrounding entities. Prior work [49, 54, 55] focuses exclusively on recognizing entities within the target sentence. However, the contexts surrounding entities of the same type are correlated, and identifying contexts that are strongly associated with entity types plays a crucial role in facilitating the generalization of pretrained language models for the NER task [47]. Neglecting these contextual correlations can lead to incorrect type predictions or entity omissions, severely impeding the adaptation of LLMs to zero-shot scenarios. For instance, as shown in Figure 1(a), the existing method [54] fails to recognize "member" and "teams," which are closely related to Person entities, in the target sentence, resulting in the omission of the entity "Atessis." To tackle this issue, we propose redefining the traditional NER task into two subtasks: recognizing named entities and identifying entity *type-related features* (TRFs [47], i.e., tokens strongly associated with entity types).

Challenge 2: Indiscriminate use of task demonstrations. To enhance task understanding and guide inference, recent zero-shot NER methods [54, 55] use both task instructions and task-specific demonstrations as input prompts for LLMs. However, these methods employ shallow strategies for demonstration retrieval, such as random sampling and k-nearest neighbors, resulting in the frequent emergence of low-quality demonstrations. For instance, as illustrated in Figure 1(b), approximately 87.33% and 76.94% of selected demonstrations do not contain any target entity types in the test sentences from the WNUT-17 and OntoNotes datasets, respectively. The indiscriminate use of unhelpful or irrelevant demonstrations substantially misleads the inference process of LLMs and degrades models' prediction abilities. To address this problem, we incorporate a self-reflection mechanism [1, 38], enabling LLMs to reflect on the helpfulness of retrieved demonstrations and selectively learn from them.

Contributions. Note that existing LLMs suffer from severe performance degradation with long input contexts [13, 21] and complex instruction following [33, 34]. Thus, it is challenging to effectively capture contextual correlations and discriminative use retrieved demonstrations through single-turn or multi-turn dialogues with LLMs. Inspired by the demonstrated complex problem-solving capabilities of multi-agent approaches [7, 52], in this paper, we present a framework, named the *cooperative multi-agent system* (CMAS) for zero-shot NER, harnessing the collective intelligence of LLM-based agents to address the challenges listed. As Figure 2 illustrates, CMAS consists of four main agents: (i) a self-annotator, (ii) a type-related feature extractor, (iii) a demonstration discriminator, and (iv) an overall predictor. First, adopting the self-improving strategy outlined in [55], CMAS employs an LLM as the self-annotator to create self-annotated data through predictions on unlabeled corpora. Then, to empower the simultaneous extraction of entities and contextual correlations, CMAS redefines the NER task into two subtasks: recognizing named entities and identifying entity type-related features within the target sentence. To achieve this, an LLM-based type-related feature extractor is developed to pinpoint words or phrases closely related to different entity types from the surrounding contexts using specialized in-context learning (ICL). Additionally, pseudo-labels for TRFs are generated using mutual information criteria, streamlining the inference process of the TRF extractor without requiring human interventions. Given the entity type features relevant to the target sentence, the demonstration discriminator integrates a self-reflection mechanism [1] to automatically assess the helpfulness of each selected demonstration in making predictions on the target test sentences. Finally, with the extracted TRFs and predicted helpfulness scores of demonstrations, the overall predictor performs inference on each incoming test sentence with a two-stage self-consistency strategy [46, 54], selectively learning from retrieved demonstrations while considering contextual correlations. Additionally, external tools, such as a syntactic structure analyzer [10], can be used to further enhance CMAS (see Section 7.1).

[2] https://catalog.ldc.upenn.edu/LDC2013T19

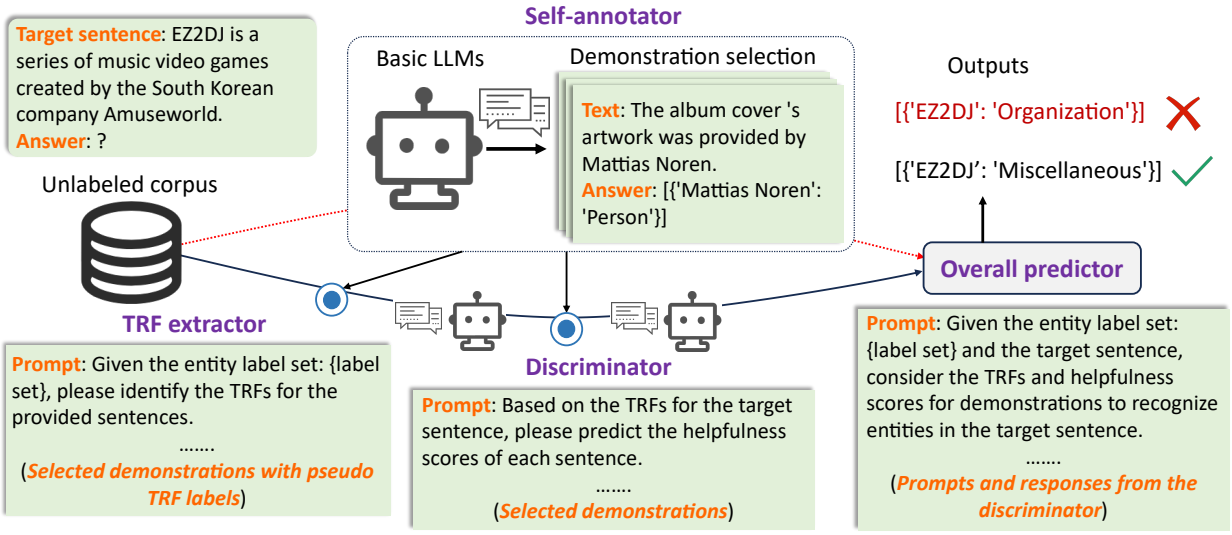


Figure 2: An overview of CMAS. The dotted red lines indicate the workflow of an existing method [55], which leads to incorrect predictions. In contrast, the solid black lines illustrate the workflow of the proposed CMAS, which consists of four key agents: (i) a self-annotator, (ii) a type-related feature extractor, (iii) a demonstration discriminator, and (iv) an overall predictor.

Our contributions are summarized as follows: (i) To the best of our knowledge, ours is the first study to design a cooperative multi-agent system for zero-shot NER that harnesses the collaborations and unique roles of multiple agents to integrate contextual correlations and the self-reflection mechanism. (ii) We redefine NER into two subtasks: recognizing named entities and identifying entity type-related features. In this way, CMAS effectively captures contextual correlations between the contexts during entity recognition, thereby reducing incorrect type predictions and entity omissions. (iii) We incorporate a self-reflection mechanism into the demonstration discriminator. By evaluating the helpfulness scores for entity extractions in target sentences, CMAS is capable of discriminately using and learning from selected demonstrations. (iv) Experimental results across six benchmarks demonstrate that our proposed CMAS achieves state-of-the-art performance on zero-shot NER and exhibits strong robustness across varying numbers of task demonstrations.

2 Related Work

2.1 Reasoning with LLMs

LLMs demonstrate strong zero-shot and few-shot reasoning capabilities, particularly when prompted to provide intermediate rationales for solving problems. Recent studies in both few-shot and zero-shot frameworks explore eliciting a chain-of-thought (CoT) process from LLMs. These studies encourage LLMs to refine their responses incrementally, enhancing the reasoning process step-by-step [15, 41, 48, 59]. Additionally, strategies like problem decomposition such as least-to-most prompting [61], break down complex problems into manageable sub-problems, addressing them sequentially. The self-consistency approach [46] involves generating a diverse array of responses from an LLM, subsequently selecting the optimal answer by averaging over these possibilities. In this paper, we focus on investigating the zero-shot reasoning ability of LLM on the NER task.

2.2 LLMs for IE

Evaluating the performance of LLMs on IE tasks is garnering significant attention [16, 18, 22, 45, 51, 60]. Wei et al. [49] propose a two-stage chatting paradigm for IE. In the first stage, ChatGPT is tasked with recognizing types of elements. In the second stage, it extracts mentions corresponding to each identified type. Han et al. [9] present a comprehensive analysis of LLMs' capabilities in IE tasks, examining aspects such as performance, evaluation criteria, robustness, and prevalent errors.

Xie et al. [54] conduct a systematic empirical study on the reasoning abilities of LLMs in IE, particularly examining performance in zero-shot NER tasks. Xie et al. [55] focus on enhancing the performance of zero-shot NER using LLMs by introducing a training-free self-improving framework that uses an unlabeled corpus to stimulate the self-learning capabilities of LLMs. Wan et al. [40] employ the chain-of-thought (CoT) approach for relation extraction (RE), using LLMs to generate intermediate rationales based on demonstrations from the training set.

2.3 LLM-based multi-agent systems

LLMs exhibit useful capabilities in reasoning and planning, aligning with human expectations for autonomous agents capable of perceiving their environments, making decisions, and taking responsive actions [8, 20, 50, 52]. Consequently, LLM-based agents are increasingly designed to understand and generate human-like instructions, enhancing complex interactions and decision-making across various contexts [19, 38, 57]. Building on the abilities of individual LLM-based agents, the concept of LLM-based multi-agent

systems has been introduced. Such systems use the collective intelligence and specialized skills of multiple agents, enabling collaborative engagement in planning, decision-making, and discussions, mirroring the cooperative nature of human teamwork in problem-solving. Recent research demonstrates the efficacy of LLM-based agents in diverse applications, including game simulation [44, 56], software development [11, 32], society simulation [28, 29], multi-robot systems [23, 58], and policy simulation [53]. Comprehensive updates on advances in LLM-based agents are detailed in recent surveys [7, 43, 52]. To the best of our knowledge, ours is the first study to develop an LLM-based cooperative multi-agent system tailored for zero-shot NER tasks.

In this paper, we focus on the zero-shot NER task. The work most closely related to ours is [49, 54, 55]. However, existing methods still face two challenging problems: (i) they overlook correlations between contexts surrounding entities, and (ii) they make indiscriminate use of task demonstrations. In our proposed CMAS, to explicitly model contextual correlations within target sentences, both named entities and TRFs are simultaneously extracted using specialized ICL. To enable controllable usage of demonstrations, a self-reflection mechanism is incorporated to automatically predict the helpfulness score of each selected demonstration for inference on the target sentences.

3 Task Formulation

To explicitly capture contextual correlations during the entity extraction process, we reinterpret the original NER task as two subtasks: recognizing named entities and identifying entity type-related features within the target sentence.

Zero-Shot NER. In this paper, we focus on the NER task in the strict zero-shot setting [55]. In this setting, no annotated data is available; instead, we only have access to an unlabeled corpus $\mathcal{D}_u$. Specifically, given an input sentence $\mathbf{x} = x_1, \ldots, x_n$ with n words from the test set $\mathcal{D}_t$, our aim is to recognize structured outputs $\mathbf{y}$ from $\mathbf{x}$, consisting of a set of (e, t) pairs. Here, e is an entity span, a sequence of tokens from $\mathbf{x}$, and t is the corresponding entity type from a predefined set $\mathcal{T}$, such as persons, location, or miscellaneous.

TRF extraction. Type-related features (TRFs), which are tokens strongly associated with entity types, are critical for the generalization of NER models [47]. Given an input sentence $\mathbf{x} \in \mathcal{D}_t$ including entity types $\mathcal{T}_x$, the goal of TRF extraction is to identify all TRFs $\mathcal{R}$ that are related to the input sentence $\mathbf{x}$ for all entity types in $\mathcal{T}_x$. Each TRF $\mathbf{w} \in \mathcal{R}$ is an m-gram span. For instance, as illustrated in Figure 1(a), "member" and "teams" are TRFs associated with the Person entity type, while "music video games" is recognized as a TRF for the Miscellaneous type.

4 CMAS: A Cooperative Multi-Agent System

In this section, we present the four main agents of the proposed CMAS as described in Figure 2: (i) a self-annotator (see Section 4.1), (ii) a type-related feature extractor (see Section 4.2), (iii) a demonstration discriminator (see Section 4.3), and (iv) an overall predictor (see Section 4.4).

First, the self-annotator uses an LLM to produce self-annotated data by making predictions on the unlabeled corpus and preliminarily retrieves demonstrations using a similarity-based strategy. Next, the type-related feature extractor automatically acquires pseudo-labels for TRFs using mutual information criteria and identifies words or phrases strongly associated with different entity types using specialized ICL. Subsequently, the demonstration discriminator incorporates a self-reflection mechanism to evaluate the helpfulness scores of each retrieved demonstration for predictions on the target input sentence. Finally, given the extracted TRFs and predicted helpfulness scores, the overall predictor performs inference on the target sentences by employing question-answering prompts and a two stage self-consistency strategy.

4.1 Self-annotator for unlabelled data

As mentioned in Section 1 and 3, we only have access to an unlabeled corpus $\mathcal{D}_u$ in zero-shot NER. Inspired by the self-improvement strategy [55], we specify an LLM-based self-annotator to guide the inference of LLMs, which first annotates the unlabeled corpus with zero-shot prompting and then preliminarily selects demonstrations from the self-annotated data for each target sentence.

Self-annotation. For each unlabeled sample $x_i \in \mathcal{D}_u$, we generate predictions using LLMs with zero-shot prompting. This process is formulated as follows:

$$\mathbf{y}_i = \arg\max_{\mathbf{y}} P_s(\mathbf{y}|\mathbf{T}_s, \mathbf{x}_i), \tag{1}$$

where $\mathbf{T}_s$ is the prompt template used for self-annotation. Prompt 1 (in the Appendix) illustrates an instance of $\mathbf{T}_s$. The predictions $y_i = \{(e_i^j, t_i^j)\}_{j=1}^{l}$ consist of pairs of entity mentions and types, where l is the number of the predicted entity mentions. P_s is the output probability of the self-annotator. To enhance the reliability of the annotations, we use self-consistency [46] and adopt a *two-stage majority voting* strategy [54]. We sample multiple responses from the model. In the first stage, we consider a candidate mention as an entity if it is present in more than half of all responses; otherwise, we discard it. In the second stage, for each mention retained from the first stage, we determine the entity type label based on the majority agreement among the responses and assign this as the final predicted label.

Preliminary demonstration selection. When a target sentence $\mathbf{x}^q$ arrives, our goal is to retrieve k relevant demonstrations $\mathbf{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{k}$ from the reliable self-annotated dataset. To achieve a better trade-off between similarity, diversity, and reliability, we employ a *diverse nearest neighbors with self-consistency ranking* strategy [55], which first retrieves K nearest neighbors based on cosine similarities between sentence representations and then selects samples with the top-k self-consistency scores.

4.2 Type-related feature extractor

To capture correlations between contexts surrounding entities, we design an LLM-based type-related feature (TRF) extractor using specialized in-context learning (ICL). We use mutual information criteria [47] to generate pseudo TRF labels $\{\mathcal{R}_i\}_{i=1}^{k}$ for self-annotated demonstrations $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{k}$, which are selected for the test input $\mathbf{x}^q$. Building on this, we apply the specialized ICL prompts to identify relevant TRFs $\mathcal{R}^q$ for $\mathbf{x}^q$.

Pseudo-label generation. To facilitate TRF extraction for the target sentence $\mathbf{x}^q$, we generate pseudo TRF labels for its self-annotated demonstrations $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^k$ using a mutual information-based method [47]. We define $\mathcal{D}_t$ as the set containing all sentences from the unlabeled corpus $\mathcal{D}_u$ where entities of the t-th type appear. The complementary set, $\mathcal{D}_u \backslash \mathcal{D}_t$, includes sentences that do not contain entities of the t-th type. To identify TRFs $\mathcal{R}^t$ associated with the t-th entity type, we apply the following filtering condition:

$$\frac{C_{\mathcal{D}_u \backslash \mathcal{D}_t}(\mathbf{w})}{C_{\mathcal{D}_t}(\mathbf{w})} \leq \rho, \quad C_{\mathcal{D}_t}(\mathbf{w}) > 0, \tag{2}$$

where $C_{\mathcal{D}_t}(\mathbf{w})$ represents the count of the m-gram $\mathbf{w}$ in $\mathcal{D}_t$, and $C_{\mathcal{D}_u \backslash \mathcal{D}_t}(\mathbf{w})$ represents its count in the rest of $\mathcal{D}_u$ excluding $\mathcal{D}_t$. The parameter ρ is an m-gram frequency ratio hyperparameter. By applying this criterion, we ensure that $\mathbf{w}$ is considered a part of the TRF set of $\mathcal{D}_t$ only if its frequency in $\mathcal{D}_t$ is significantly higher than its frequency in other sets ($\mathcal{D}_u \backslash \mathcal{D}_t$). Given the smaller size of $\mathcal{D}_t$ relative to $\mathcal{D}_u \backslash \mathcal{D}_t$, we select $\rho \geq 1$ but avoid excessively high values to include features associated with $\mathcal{D}_t$ and potentially relevant to other entity types within the TRF set of $\mathcal{D}_t$. Based on this, for every self-annotated demonstration $\mathbf{x}_i \in \mathcal{S}$, we compute the Euclidean distance between BERT-based embeddings of each extracted TRF $\mathbf{w}$ and each token in $\mathbf{x}_i$, selecting the top-5 features as pseudo TRF labels $\mathcal{R}_i$ of $\mathbf{x}_i$.

TRF Extraction. Given the target sentence $\mathbf{x}^q$ and its corresponding demonstrations $\mathcal{S}_d = \{\mathbf{x}_i, \mathbf{y}_i, \mathcal{R}_i\}_{i=1}^k$ with pseudo-labels, we construct specialized ICL prompts to facilitate the identification of relevant TRFs for $\mathbf{x}^q$. The inference process is formulated as:

$$\mathcal{R}^q = \arg\max_{\mathcal{R}} P_e(\mathcal{R}|\mathbf{T}_e, \mathcal{S}_d, \mathbf{x}^q), \tag{3}$$

where $\mathbf{T}_e$ represents the specialized ICL prompt template. Prompt 2 (in the Appendix) provides a detailed instance of $\mathbf{T}_e$. $P_e(\cdot)$ represents the output probability of the TRF extractor.

4.3 Demonstration discriminator

As mentioned in Section 1, demonstrations retrieved using shallow similarity-based strategies can be highly irrelevant to target sentences, severely misleading the predictions of LLMs. To address this issue, we employ an LLM-based demonstration discriminator with a self-reflection mechanism [1, 38] to automatically evaluate the helpfulness of each initially selected demonstration for making predictions on the target test sentences. To achieve this, we consider the TRFs of both the demonstrations and the target sentence, extracted by the TRF extractor (see Section 4.2), as well as the self-labeled entity labels from the self-annotator (see Section 4.1). The prompts used for helpfulness score prediction are shown in detail in Prompt 3 (in the Appendix). Formally, given demonstrations $\mathcal{S}_d = \{\mathbf{x}_i, \mathbf{y}_i, \mathcal{R}_i\}_{i=1}^k$ selected for the target sentence $\mathbf{x}^q$ with extracted TRFs $\mathcal{R}^q$, the corresponding helpfulness scores $\{h_i\}_{i=1}^k$ are predicted by

$$h_i = \arg\max_{h} P_d(h|\mathbf{T}_d, \mathcal{S}_d, \mathcal{R}^q, \mathbf{x}_i, \mathbf{x}^q), \tag{4}$$

where $\mathbf{T}_d$ denotes the prompt template used in the demonstration discriminator. $P_d(\cdot)$ denotes the output probability of the demonstration discriminator.

4.4 Overall predictor

Finally, given the extracted TRFs and predicted helpfulness scores, we establish an LLM-based overall predictor to conduct inference on the target sentences. Let $\mathcal{S}_o = \{\mathbf{x}_i, \mathbf{y}_i, h_i, \mathcal{R}_i\}_{i=1}^k$ represent the self-annotated demonstrations selected for the test input $\mathbf{x}^q$, along with the corresponding helpfulness scores $\{h_i\}_{i=1}^k$ and TRFs $\{\mathcal{R}_i\}_{i=1}^k$. As Prompt 4 (in the Appendix) shows, CMAS conducts overall predictions on $\mathbf{x}^q$ by integrating the dialogue in the demonstration discriminator and constructing a question-answering prompt template $\mathbf{T}_o$. The overall prediction is obtained by

$$\mathbf{y}^q = \arg\max_{\mathbf{y}} P_o(\mathbf{y}|\mathbf{T}_o, \mathcal{S}_o, \mathbf{x}^q), \tag{5}$$

where $P_o(\cdot)$ denotes the output probability of the overall predictor. Similar to the self-annotation process (see Section 4.1), to improve the reliability and consistency of the final results, we sample multiple responses from LLMs and adopt a two-stage self-consistency strategy.

Now that we have described the four specialized agents that make up the core of CMAS, we recall the coordinated workflow of CMAS, as illustrated in Figure 2. To start, the self-annotator incorporates a self-improvement strategy and employs an LLM to generate self-annotated data from an unlabeled corpus $\mathcal{D}_u$, and preliminarily retrieves reliable demonstrations $\mathbf{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^k$ for each test input $\mathbf{x}^q$. Next, the type-related feature extractor uses mutual information criteria to derive pseudo TRF labels $\{\mathcal{R}_i\}_{i=1}^k$ for the self-annotated demonstrations and facilitates identifying relevant contextual correlations $\mathcal{R}^q$ for $\mathbf{x}^q$ using specialized ICL. Furthermore, the demonstration discriminator, considering the TRFs of both the target sentence and its self-annotated demonstrations, applies a self-reflection mechanism to automatically assess the helpfulness scores $\{h_i\}_{i=1}^k$ of selected demonstrations in making predictions for each target sentence $\mathbf{x}^q$. Finally, the overall predictor constructs a question-answering prompt template $\mathbf{T}_o$, synthesizing TRFs and helpfulness scores, to obtain the final prediction $\mathbf{y}^q$ for each target sentence $\mathbf{x}^q$. Through their specialized abilities and communications, the designed agents work collaboratively to enhance both the effectiveness and generalizability of our proposed CMAS for zero-shot and few-shot NER tasks (see Section 6 and 7).

5 Experiments

Research questions. We aim to answer the following research questions: (RQ1) Does CMAS outperform state-of-the-art methods on the zero-shot NER task? (See Section 6.1) (RQ2) Can CMAS be generalized to the few-shot setting? (See Section 6.2)

Datasets. In our experiments, we evaluate CMAS on both general-domain and domain-specific datasets. Detailed statistics of the datasets used are shown in Table 5 (in the Appendix). For our general-domain experiments, we consider four commonly used benchmarks, namely the CoNLL03 [36], WikiGold [2], WNUT-17 [4], and OntoNotes[3] datasets. For our domain-specific experiments, we evaluate CMAS on the GENIA [25] and BioNLP11 [31] datasets in the biomedical domain.

[3] https://catalog.ldc.upenn.edu/LDC2013T19

For zero-shot NER, to keep API usage costs under control, we randomly sample 300 examples from the test set three times and calculate the average performance. An exception is made for WikiGold, which has a test set of only 247 examples. Additionally, we randomly sample 500 examples from the training set as the unlabeled corpus. In the few-shot settings, we examine configurations including 0-shot, 3-shot, 5-shot, and 10-shot, where "shot" refers to the number of demonstrations with gold labels provided to LLMs (see Section 6.2). For each setting, we randomly sample three sets of demonstrations and compute the average results.

Baselines For the zero-shot and few-shot NER tasks, we compare CMAS with the following baselines: (i) **Vanilla** [54, 55] employs a straightforward and common prompting strategy that directly asks LLMs to extract entity labels from input texts. (ii) **ChatIE** [49] transforms the zero-shot NER task into a multi-turn question-answering problem using a two-stage framework. (iii) **Decomposed-QA** [54] breaks down the zero-shot NER task into a series of simpler subproblems by labels and follows a decomposed-question-answering paradigm, where the model extracts entities of only one label at a time. (iv) Based on Decomposed-QA, **SALLM** [54] further adopts syntactic augmentation to stimulate the model's intermediate reasoning in two ways, including syntactic prompting and tool augmentation. (v) **SILLM** [55] applies a self-improving framework, which uses unlabeled corpora to stimulate the self-learning abilities of LLMs in zero-shot NER.

In our experiments, we evaluate all the aforementioned baselines for the zero-shot NER task. For few-shot NER, we assess SILLM [55] and Vanilla [54, 55] since ChatIE [49], Decomposed-QA [54], and SALLM [54] do not incorporate task demonstrations in their prompt templates. Additionally, we report the highest results obtained from the model variants of SALLM [54] and SILLM [55] in our experiments. For fair comparisons and cost savings, we employ GPT-3.5 (specifically, the `gpt-3.5-turbo-0125` model[4]) as the LLM backbone for all baselines and agents in CMAS. We use the `text-embedding-ada-002` model,[5] a text-embedding model from OpenAI, to obtain sentence representations. We access OpenAI models using the official API.

5.1 Evaluation metrics

Following previous work [54, 55], we conduct our evaluation of zero-shot and few-shot NER tasks using only complete matching and employing the micro F1-score to assess the NER task. We consider a prediction correct only when both the boundary and the type of the predicted entity exactly match those of the true entity.

5.2 Implementation details

Following Xie et al. [55] and Wang et al. [42], we set the number of nearest neighbors $K = 50$ during self-annotation and the number of task demonstrations $k = 16$. For self-consistency scores, we set the temperature to 0.7 and sample 5 answers. We set the frequency ratio hyperparameter ρ to 3 for all experiments and only consider 1-gram texts for simplicity.

[4] https://platform.openai.com/docs/models/gpt-3-5-turbo
[5] https://platform.openai.com/docs/models/embeddings

6 Experimental Results

To answer RQ1 and RQ2, we assess the performance of CMAS on both zero-shot and few-shot NER tasks.

6.1 Results on zero-shot NER

We turn to RQ1. Table 1 shows the experimental results on both general-domain and domain-specific datasets. To ensure fair comparisons, we reproduce all the baseline models using the `gpt-3.5-turbo-0125` model, as they are implemented with different versions of GPT-3.5 in the original papers. We have the following observations: (i) Zero-shot NER is challenging, and baseline models struggle to achieve an F1-score above 60% on most of the datasets. For instance, ChatIE only obtains F1-scores of 37.46% and 29.00% on the WNUT-17 and OntoNotes datasets, respectively. (ii) CMAS achieves the highest F1-scores across all datasets, indicating its superior performance. For instance, CMAS attains F1-scores of 76.23% and 60.51% on the WikiGold and BioNLP11 datasets, respectively. (iii) CMAS significantly outperforms the previous state-of-the-art baselines across all datasets. For example, CMAS achieves improvements of 13.21% and 4.49% over the best-performing baselines on the WNUT-17 and GENIA datasets, respectively.

In summary, CMAS demonstrates its effectiveness in recognizing named entities in a strict zero-shot setting. The identification of contextual correlations and the evaluation of helpfulness scores for task demonstrations are beneficial for zero-shot NER.

6.2 Results on few-shot NER

To investigate the effectiveness of CMAS on the few-shot setting, we turn to RQ2. In our experiments, we evaluate CMAS and the baselines in 0-shot, 3-shot, 5-shot, and 10-shot settings, where micro F1-scores are reported. ChatIE, Decomposed-QA, and SALLM are excluded because incorporating gold demonstrations into their prompt templates is non-trivial and beyond the scope of this paper.

Based on Figure 3, we arrive at the following conclusions: (i) Increasing the number of demonstrations with gold labels does not necessarily enhance the prediction performance of LLMs. For example, as the number of demonstrations with gold labels increases from 0 to 10, the F1-score of the Vanilla model significantly drops from 40.10% to 27.10% on the WNUT-17 dataset. This decline may be due to the random selection of demonstrations, which can be highly irrelevant to the target sentence and severely misguide the inference process of LLMs. (ii) CMAS achieves the highest F1-scores and consistently outperforms the state-of-the-art baselines across all few-shot settings, demonstrating its effectiveness and robustness. For example, CMAS exhibits an average improvement of 19.51% and 13.10% over SILLM on the WNUT-17 and GENIA datasets, respectively.

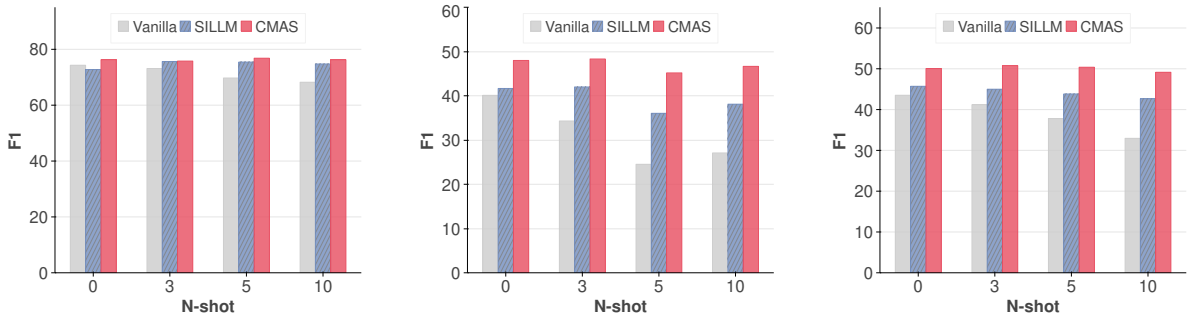
In summary, our proposed CMAS not only effectively extracts entities in the strict zero-shot setting but also achieves the highest F1-scores across all few-shot settings while maintaining robustness to irrelevant demonstrations.

7 Analysis

Now that we have addressed our research questions, we take a closer look at CMAS to analyze its performance and generalizability. We examine the contributions of the type-related feature

Table 1: Zero-shot NER results (F1) on both general-domain and domain-specific datasets. Numbers in bold are the highest results for the corresponding dataset, while numbers underlined represent the second-best results. Significant improvements against the best-performing baseline for each dataset are marked with $*$ (t-test, $p < 0.05$).

Model	Datasets					
	CoNLL03	WikiGold	WNUT-17	OntoNotes	GENIA	BioNLP11
Vanilla [54, 55]	72.54	74.27	40.10	45.09	43.47	53.92
ChatIE [49]	50.13	56.78	37.46	29.00	47.85	45.56
Decomposed-QA [54]	52.61	64.05	42.38	35.96	34.03	57.26
SALLM [54]	68.97	72.14	38.66	44.53	42.33	55.06
SILLM [55]	72.96	72.72	41.65	45.34	45.66	44.99
CMAS (ours)	**76.43***	**76.23***	**47.98***	**46.23***	**50.00***	**60.51***



(a) WikiGold. (b) WNUT-17. (c) GENIA.

Figure 3: Few-shot NER results (F1) on WikiGold, WNUT-17, and GENIA.

extractor and the demonstration discriminator to its effectiveness (see Section 7.1), investigate its generalizability to different LLM backbones (see Section 7.2) and varying numbers of task demonstrations (see Section A.3), and assess its capability in error correction (see Section 7.3).

7.1 Ablation studies

To study the individual contributions of each component to CMAS's performance, we conduct ablation studies on the WikiGold, WNUT-17, and GENIA datasets. The results are presented in Table 2.

Given that the demonstration discriminator relies on entity type-related information from the TRF extractor, it is not feasible to independently remove the TRF extractor. When we ablate only the demonstration discriminator ('- Discriminator'), the overall predictor incorporates only TRF for retrieved demonstrations and target sentences. This exclusion results in a significant drop in CMAS's performance across all three datasets. For instance, CMAS achieves 3.34% and 5.59% higher F1-scores on the WikiGold and GENIA datasets, respectively, compared to its model variant without the demonstration discriminator. These findings highlight the crucial role of evaluating the usefulness of retrieved demonstrations in making predictions. In scenarios where both the demonstration discriminator and the TRF extractor are ablated ('- TRF Extractor'), CMAS reverts to the baseline model, SILLM. The results indicate that identifying contextual correlations surrounding entities considerably enhances SILLM's performance. In summary, both the demonstration discriminator and the TRF extractor contribute markedly to CMAS's performance improvements over the baselines in the zero-shot NER task.

Furthermore, similar to SALLM, CMAS is readily adaptable for augmentation with external syntactic tools. Following Xie et al. [54], we obtain four types of syntactic information (i.e., word segmentation, POS tags, constituency trees, and dependency trees) via a parsing tool [10] and integrate the syntactic information into the overall predictor of CMAS using a combination of tool augmentation and syntactic prompting strategies. As shown in Table 2, the inclusion of dependency tree information improves CMAS's performance by 2.52% and 2.94% on WNUT-17 and GENIA, respectively. These results demonstrate that the integration of appropriate external tools further enhances the performance of CMAS.

7.2 Influence of different LLM backbones

To explore the impact of different LLM backbones, we evaluate CMAS and baseline models using the latest LLMs, including GPT (`gpt-3.5-turbo-0125`), Llama (`Meta-Llama-3-8B-Instruct`[6]), and Qwen (`Qwen2.5-7B-Instruct`[7]). Table 3 illustrates the zero-shot NER performance on the WNUT-17 and GENIA datasets. We exclude the performance of ChatIE and Decomposed-QA, as their F1-scores with Qwen and Llama backbones are considerably lower than other baselines. As Table 3 shows, CMAS achieves the highest F1-scores when using GPT as the backbone model. Additionally,

[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[7] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Table 2: Ablation studies (F1) on WikiGold, WNUT-17, and GENIA.

Model	Datasets		
	WikiGold	WNUT-17	GENIA
Vanilla [54, 55]	74.27	40.10	43.47
ChatIE [49]	56.78	37.46	47.85
Decomposed-QA [54]	64.05	42.38	34.03
SALLM [54]	72.14	38.66	42.33
CMAS (ours)	**76.23**	**47.98**	**50.00**
- Discriminator	73.76	45.44	48.41
- TRF extractor	72.72	41.65	45.66
External tool augmentation			
Word segmentation	**76.92**	47.63	49.22
POS tag	76.14	48.11	49.76
Constituency tree	75.71	47.44	49.64
Dependency tree	76.27	**49.19**	**51.47**

Table 3: Influence of different LLM backbones (F1) on WNUT-17 and GENIA. Numbers in bold are the highest results for the corresponding dataset, while numbers underlined represent the second-best results. Significant improvements against the best-performing baseline for each dataset are marked with $*$ (t-test, $p < 0.05$).

Model	WNUT-17			GENIA		
	GPT	Llama	Qwen	GPT	Llama	Qwen
Vanilla [54, 55]	40.10	34.88	34.93	43.47	15.36	9.97
SALLM [54]	38.66	40.95	41.50	42.33	36.23	19.13
SILLM [55]	41.65	22.43	36.23	45.66	28.13	33.80
CMAS (ours)	**47.98***	**42.36***	**44.62***	**50.00***	**45.68***	**36.12***

CMAS consistently outperforms the baselines across various LLM backbones, demonstrating its superiority and generalizability.

7.3 Error analysis

To investigate CMAS's error correction capabilities, we conduct an analysis of the following errors on the WNUT-17 dataset:

- **Type errors**: (i) **OOD types** are predicted entity types not in the predefined label set; (ii) **Wrong types** are predicted entity types incorrect but in the predefined label set.
- **Boundary errors**: (i) **Contain gold** are incorrectly predicted mentions that contain gold mentions; (ii) **Contained by gold** are incorrectly predicted mentions that are contained by gold mentions; (iii) **Overlap with gold** are incorrectly predicted mentions that do not fit the above situations but still overlap with gold mentions.
- **Completely-Os** are incorrectly predicted mentions that do not coincide with any of the three boundary situations associated with gold mentions.
- **OOD mentions** are predicted mentions that do not appear in the input text.
- **Omitted mentions** are entity mentions that models fail to identify.

Figure 5 (in the Appendix) visualizes the percentages of error types. The majority error types are *overlap with gold* and *ommited mentions*, which account for 72.30% of all errors. These errors may result from incomplete annotations or predictions influenced by the prior knowledge of LLMs. Table 4 (in the Appendix) summarizes the statistics of error types. With the implementation of the proposed type-related feature extractor and demonstration discriminator, CMAS significantly reduces the total number of errors by 30.60% and 74.60% compared to state-of-the-art baselines SALLM and SILLM, respectively, demonstrating its remarkable effectiveness in error correction.

8 Conclusions

We have focused on named entity recognition in the strict zero-shot setting, where no annotated data is available. Previous approaches still encounter two key challenges: they overlook contextual correlations and use task demonstrations indiscriminately, both of which impede the inference process. To tackle these challenges, we have introduced a new framework, *cooperative multi-agent system* (CMAS), which uses the collective intelligence and specialized capabilities of agents. CMAS explicitly captures correlations between contexts surrounding entities by decomposing the task into recognizing named entities and identifying entity type-related features. To enable controllable use of demonstrations, a demonstration discriminator is established to incorporate a self-reflection mechanism, automatically evaluating helpfulness scores for the target sentence. Experiments on six benchmarks, spanning domain-specific and general datasets, show that CMAS significantly improves zero-shot NER performance and effectively corrects various errors.

Expanding CMAS to support open NER tasks would be a valuable direction for future research. Future work also includes developing interactive prompt designs, such as multi-turn question answering, to enable LLM-based agents to iteratively refine or assess responses.

Acknowledgments

This work was supported by the National Key R&D Program of China with grant No. 2022YFC3303004, the Natural Science Foundation of China (62372275, 62272274, 62202271, T2293773, 62102234, 62072279), the Natural Science Foundation of Shandong Province (ZR2022QF004, ZR2021QF129), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), the Fundamental Research Funds of Shandong University, the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organization for Scientific Research, https://hybrid-intelligence-centre.nl, project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), and the FINDHR (Fairness and Intersectional Non-Discrimination in Human Recommendation) project that received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101070212. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *ICLR*.

[2] Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named Entity Recognition in Wikipedia. In *PWNLP@IJCNLP*. 10–18.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.* 24 (2023), 240:1–240:113.

[4] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In *NUT@EMNLP*. 140–147.

[5] Yixing Fan, Jiafeng Guo, Xinyu Ma, Ruqing Zhang, Yanyan Lan, and Xueqi Cheng. 2021. A Linguistic Study on Relevance Modeling in Information Retrieval. In *WWW*. 1053–1064.

[6] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named Entity Recognition in Query. In *SIGIR*. 267–274.

[7] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. *CoRR* abs/2402.01680 (2024).

[8] Taicheng Guo, Kehan Guo, Bozhao Nan, Zhengwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2023. What Indeed Can GPT Models Do in Chemistry? A Comprehensive Benchmark on Eight Tasks. *CoRR* abs/2305.18365 (2023).

[9] Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is Information Extraction Solved by ChatGPT? An Analysis of Performance, Evaluation Criteria, Robustness and Errors. *CoRR* abs/2305.14450 (2023).

[10] Han He and Jinho D. Choi. 2021. The Stem Cell Hypothesis: Dilemma behind Multi-Task Learning with Transformer Encoders. In *EMNLP*. 5555–5577.

[11] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *ICLR*.

[12] Tomoki Ito and Shun Nakagawa. 2024. Tender Document Analyzer with the Combination of Supervised Learning and LLM-based Improver. In *WWW*. 995–998.

[13] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. LLM Maybe LongLM: SelfExtend LLM Context Window Without Tuning. In *ICML*.

[14] Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting Asymmetry for Synthetic Training Data Generation: SynthIE and the Case of Information Extraction. In *EMNLP*. 1555–1574.

[15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *NeurIPS*.

[16] Md. Tahmid Rahman Laskar, M. Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. In *ACL*. 431–469.

[17] Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. LIQUID: A Framework for List Question Answering Dataset Generation. In *AAAI*. 13014–13024.

[18] Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness. *CoRR* abs/2304.11633 (2023).

[19] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. In *EMNLP*. 3102–3116.

[20] Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kalyan. 2023. Let GPT be a Math Tutor: Teaching Math Word Problem Solvers with Customized Exercise Generation. In *EMNLP*. 14384–14396.

[21] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Trans. Assoc. Comput. Linguistics* 12 (2024), 157–173.

[22] Mingyu Derek Ma, Xiaoxuan Wang, Po-Nien Kung, P. Jeffrey Brantingham, Nanyun Peng, and Wei Wang. 2024. STAR: Boosting Low-Resource Information Extraction by Structure-to-Text Data Generation with Large Language Models. In *AAAI*. 18751–18759.

[23] Zhao Mandi, Shreeya Jain, and Shuran Song. 2023. RoCo: Dialectic Multi-Robot Collaboration with Large Language Models. *CoRR* abs/2307.04738 (2023).

[24] AI Meta. 2024. Introducing Meta Llama 3: The Most Capable Openly Available LLM to Date. *Meta AI* (2024).

[25] Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, Hideki Mima, and Junichi Tsujii. 2002. The GENIA Corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain. In *HLT*. 73–77.

[26] OpenAI. 2022. Introducing ChatGPT. (2022).

[27] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023).

[28] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *UIST*. 2:1–2:22.

[29] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. In *UIST*. 74:1–74:18.

[30] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-Supervised Sequence Tagging with Bidirectional Language Models. In *ACL*. 1756–1765.

[31] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Daniel E. Sullivan, Chunhong Mao, Chunxia Wang, Bruno W. S. Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011. *BMC Bioinform.* 13, S-11 (2012), S2.

[32] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative Agents for Software Development. *CoRR* abs/2307.07924 (2023).

[33] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating Instruction Following Ability in Large Language Models. *CoRR* abs/2401.03601 (2024).

[34] Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Guanglu Wan, Xunliang Cai, and Le Sun. 2024. Learning or Self-aligning? Rethinking Instruction Fine-tuning. In *ACL*. 6090–6105.

[35] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. *CoRR* abs/2310.03668 (2023).

[36] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *CoNLL*. 142–147.

[37] Maximilian Schmidt, Andrea Bartezzaghi, and Ngoc Thang Vu. 2024. Prompting-based Synthetic Data Generation for Few-Shot Question Answering. In *LREC/COLING*. 13168–13178.

[38] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *NeurIPS*.

[39] Fábio Souza, Rodrigo Frassetto Nogueira, and Roberto de Alencar Lotufo. 2019. Portuguese Named Entity Recognition using BERT-CRF. *CoRR* abs/1909.10649 (2019).

[40] Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. GPT-RE: In-context Learning for Relation Extraction using Large Language Models. *arXiv preprint arXiv:2305.02105* (2023).

[41] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. *arXiv preprint arXiv:2212.10001* (2022).

[42] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. August 2021. Generalizing to Unseen Domains: A Survey on Domain Generalization. In *IJCAI*. 4627–4635.

[43] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A Survey on Large Language Model based Autonomous Agents. *Frontiers Comput. Sci.* 18, 6 (2024), 186345.

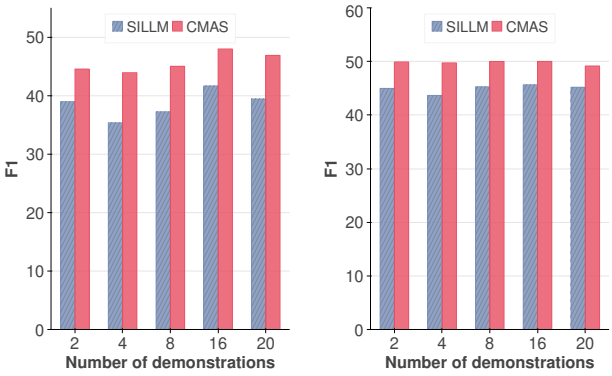
[44] Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2023. Avalon's Game of Thoughts: Battle Against Deception through Recursive Contemplation. *CoRR* abs/2310.01320 (2023).

[45] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. GPT-NER: Named Entity Recognition via Large Language Models. *CoRR* abs/2304.10428 (2023).

[46] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *ICLR*.

[47] Zihan Wang, Ziqi Zhao, Zhumin Chen, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2023. Generalizing Few-Shot Named Entity Recognizers to Unseen Domains with Type-Related Features. In *EMNLP*. 2228–2240.

[48] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large



(a) WNUT-17. (b) GENIA.

Figure 4: Influence of task demonstration amount (F1) on WNUT-17 and GENIA.

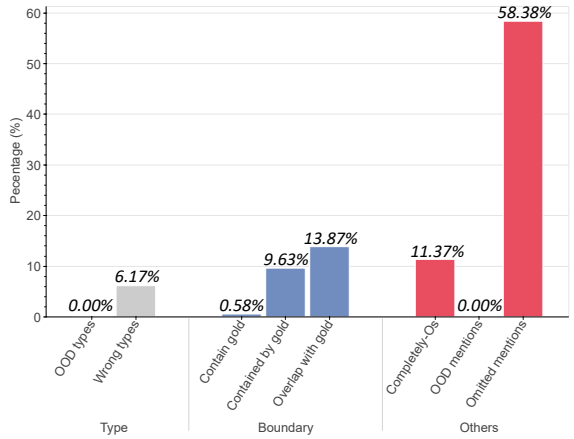


Figure 5: Percentages of different error types on GENIA using CMAS.

Language Models. *CoRR* abs/2201.11903 (2022).
[49] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023. Zero-Shot Information Extraction via Chatting with ChatGPT. *CoRR* abs/2302.10205 (2023).
[50] Michael J. Wooldridge and Nicholas R. Jennings. 1995. Intelligent Agents: Theory and Practice. *Knowl. Eng. Rev.* 10, 2 (1995), 115–152.
[51] Chenxiao Wu, Wenjun Ke, Peng Wang, Zhizhao Luo, Guozheng Li, and Wanyi Chen. 2024. ConsistNER: Towards Instructive NER Demonstrations for LLMs with the Consistency of Ontology and Context. In *AAAI*. 19234–19242.
[52] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *CoRR* abs/2309.07864 (2023).
[53] Bushi Xiao, Ziyuan Yin, and Zixuan Shan. 2023. Simulating Public Administration Crisis: A Novel Generative Agent-Based Simulation System to Lower Technology Barriers in Social Science Research. *CoRR* abs/2311.06957 (2023).
[54] Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical Study of Zero-Shot NER with ChatGPT. In *EMNLP*. 7935–7956.
[55] Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. Self-Improving for Zero-Shot Named Entity Recognition with Large Language Models. In *NAACL*. 583–593.
[56] Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2023. Language Agents with Reinforcement Learning for Strategic Play in the Werewolf Game. *CoRR* abs/2310.18940 (2023).
[57] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *NeurIPS*.
[58] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. 2023. Building Cooperative Embodied Agents Modularly with Large Language Models. *CoRR* abs/2307.02485 (2023).
[59] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *ICLR*.
[60] Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. 2024. LinkNER: Linking Local Named Entity Recognition Models to Large Language Models using Uncertainty. In *WWW*. 4047–4058.
[61] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *ICLR*.
[62] 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 *ICLR*.

A Appendix

Table 4: Numbers of different error types on GENIA. Numbers in bold denote the best results for the corresponding error type, i.e., the least errors, while numbers underlined represent the second-best results.

Error Types		ChatIE	SALLM	SILLM	CMAS
Type	OOD types	91	2	**0**	**0**
	Wrong types	52	52	123	**32**
Boundary	Contain gold	34	7	10	**3**
	Contained by gold	**42**	101	240	50
	Overlap with gold	145	135	318	**72**
Completely-Os		122	109	241	**59**
OOD mentions		2	**0**	**0**	**0**
Omitted mentions		422	342	1,109	**303**
Total		910	748	2,041	**519**

A.1 Prompts used in CMAS

Prompt 1, 2, 3, and 4 show prompts used in the self-annotator, TRF extractor, demonstration discriminator, and overall predictor, respectively.

A.2 Statistics of the datasets used

Table 5 demonstrates the detailed statistics of the dataset used in our experiments.

A.3 Influence of task demonstration amount

To assess the influence of the number of task demonstrations, we evaluate CMAS and SILLM with varying numbers of task demonstrations, ranging from 2 to 20. Figure 4 details the zero-shot NER performance on the WNUT-17 and GENIA datasets. It is important to note that other baselines are excluded from this analysis as they do not incorporate task demonstrations in their prompt templates. The results in Figure 4 indicate that CMAS consistently outperforms SILLM across various numbers of task demonstrations on both datasets. Specifically, CMAS achieves an average F1-score improvement of 18.70% and 10.72% over SILLM on the WNUT-17 and

GENIA datasets, respectively. This is, CMAS is able to effectively discriminate against irrelevant task demonstrations, maintaining robustness across different numbers of task demonstrations.

A.4 Percentages of different error types

Table 4 and Figure 5 present the statistics and percentages of error types on GENIA using CMAS, respectively.

Prompt 1: Prompts used for the self-annotator.

Prompts used for the self-annotator
Given entity label set: ['Organization', 'Person', 'Location', 'Miscellaneous'], please recognize the named entities in the given text. Provide the answer in the following JSON format: [{'Entity Name': 'Entity Label'}]. If there is no entity in the text, return the following empty list: []. Text: The album cover's artwork was provided by Mattias Noren. Answer:

Prompt 2: Prompts used for the TRF extractor.

Prompts used for the TRF extractor
Here, we provide some example sentences with the corresponding TRFs. TRFs mean type-related features, which are tokens that are strongly associated with the entity types and relevant to these sentences. Given entity label set: ['Organization', 'Person', 'Location', 'Miscellaneous'], please identify the TRFs for the target sentences. Provide the answer in the following list format: ['TRF1', 'TRF2', ...]. Text: His parents were encouraged by a friend to develop the child's musical talents and he studied classical piano in the United States. TRF set: ['father', 'songs', 'player', 'United States', 'French'] ..... (Selected demonstrations with pseudo TRF labels) ..... Target sentence: UK Edition came with the OSC-DIS video, and most of the tracks were re-engineered. TRF set:

Prompt 3: Prompts used for the demonstration discriminator.

Prompts used for the demonstration discriminator
Here, we provide some example sentences and the corresponding entity labels and TRFs. TRFs mean type-related features, which are tokens that are strongly associated with the entity types and relevant to these sentences. Given entity label set: ['Organization', 'Person', 'Location', 'Miscellaneous'], target sentence: 'UK Edition came with the OSC-DIS video, and most of the tracks were re-engineered.' and its TRF set:['video', 'tracks']. Please predict the helpfulness scores of each sentence, which indicates the degree to which providing the current example can aid in extracting named entities from the target sentence. The score ranges from 1 to 5, with 1 being the least helpful and 5 being the most helpful. Provide answer in the following JSON format: [{'sentence id': 'helpfulness score'}]. Text: His parents were encouraged by a friend to develop the child's musical talents and he studied classical piano in the United States. TRF set: ['father', 'songs', 'player', 'United States', 'French'] Entity labels: [{'United States': 'Location'}] ..... (Selected demonstrations with pseudo TRF labels and entity labels) ..... Answer:

Table 5: Statistics of the datasets used. The training set is formed by combining the original training and development sets.

Dataset	CoNLL03	WikiGold	WNUT-17	OntoNotes	GENIA	BioNLP11
#Train	14,382	1,422	4,403	68,452	16,692	3,217
#Test	3,453	274	1,287	8,262	1,854	1,961

Prompt 4: Prompts used for the overall predictor.

Prompts used for the overall predictor
…… (The corresponding prompt for the demonstration discriminator and its response) …… Given entity label set: [‘Organization’, ‘Person’, ‘Location’, ‘Miscellaneous’], please consider the TRFs and helpfulness scores for the above sentences to recognize the named entities in the target sentence. Provide the answer in the following JSON format: [{ ‘Entity Name’: ‘Entity Label’}]. If there is no entity in the text, return the following empty list: []. Target sentence: UK Edition came with the OSC-DIS video, and most of the tracks were re-engineered. Answer: