MAF-IE: Multi-Agent Finetuning for Zero-Shot Information Extraction

Anonymous ACL submission

Abstract

Large language models (LLMs) excel at text generation and reasoning but struggle with producing structured output while maintaining accuracy in zero-shot information extraction (IE). Recent studies have explored multi-agent frameworks to enhance LLMs' capabilities, but these efforts primarily target general reasoning and fail to address key structured IE challenges such as boundary ambiguity and cross-type semantic conflicts. In this work, we propose MAF-IE, a multi-agent finetuning framework that combines specialization and collaborative training to improve both the accuracy and efficiency of multi-agent systems for IE. Specifically, we introduce a type-specified multi-agent collaboration framework to generate high-quality pseudo-labeled data. Based on the generated data, we design a novel contrastive data selection strategy to finetune multiple LLMs on dialogue trajectories, enabling the model to better learn from both correct and incorrect predictions, enhancing task-specific feature learning. Combined with a simple majority voting strategy, the finetuned models achieve comparable performance to multi-agent LLMs while significantly reducing inference costs. Extensive experiments on seven datasets across five tasks, spanning coarse- and fine-grained settings at both sentence and document levels, demonstrate MAF-IE significantly outperforms zero-shot IE baselines.

1 Introduction

Information extraction (IE) converts unstructured or semi-structured text into structured representations (Li et al., 2023c; Lu et al., 2022). Traditional supervised IE methods adapt pre-trained language models to labeled datasets with supervision signals (Devlin et al., 2019; Zhuang et al., 2021), but they rely on costly annotations and struggle to generalize to low-resource or evolving domains. To address these limitations, zero-shot paradigms have emerged as a promising alternative by leveraging LLMs' strong language understanding capabilities acquired through extensive pre-training (Xie et al., 2023; Wang et al., 2023a). However, a single LLM under zero-shot often achieves suboptimal results. For instance, directly prompting GPT-3.5 yields only 45% F1 on CoNLL03 and 34% on OntoNotes4 (Li et al., 2024b), highlighting a significant gap between zero-shot methods and reliable structured extraction.

To bridge this gap, recent strategies utilize advanced models like GPT-4 (OpenAI, 2024a) to generate synthetic supervision (Heng et al., 2024; Ye et al., 2024), but their effectiveness is bounded by model capability and constrained by heavy computational and legal requirements. Another promising direction employs multi-agent frameworks, enabling multiple LLMs to collaborate through voting (Wang et al., 2023c), debate (Chen et al., 2024) or decision-making (Sun et al., 2025). These systems promote diverse reasoning paths (Du et al., 2023), critique each other's outputs (Chan et al., 2023) and aggregate complementary predictions into a final output to address a single model's limitations (Pham et al., 2024; Zhao et al., 2025).

However, existing multi-agent frameworks face critical challenges that hinder their direct applicability to diverse IE tasks, including limited task-specific adaptation, poor scalability caused by coordination overhead, and insufficient flexibility to accommodate varying IE task requirements. The fundamental issue lies in their high computational costs and low efficiency, making them impractical for time-sensitive or large-scale applications. Ideally, these benefits could be achieved by a single model that performs direct inference with both high efficiency and practicality.

In this paper, we propose MAF-IE, a novel multi-agent finetuning framework that distills collaborative strengths into a set of finetuned models. Our method is specifically designed for IE,

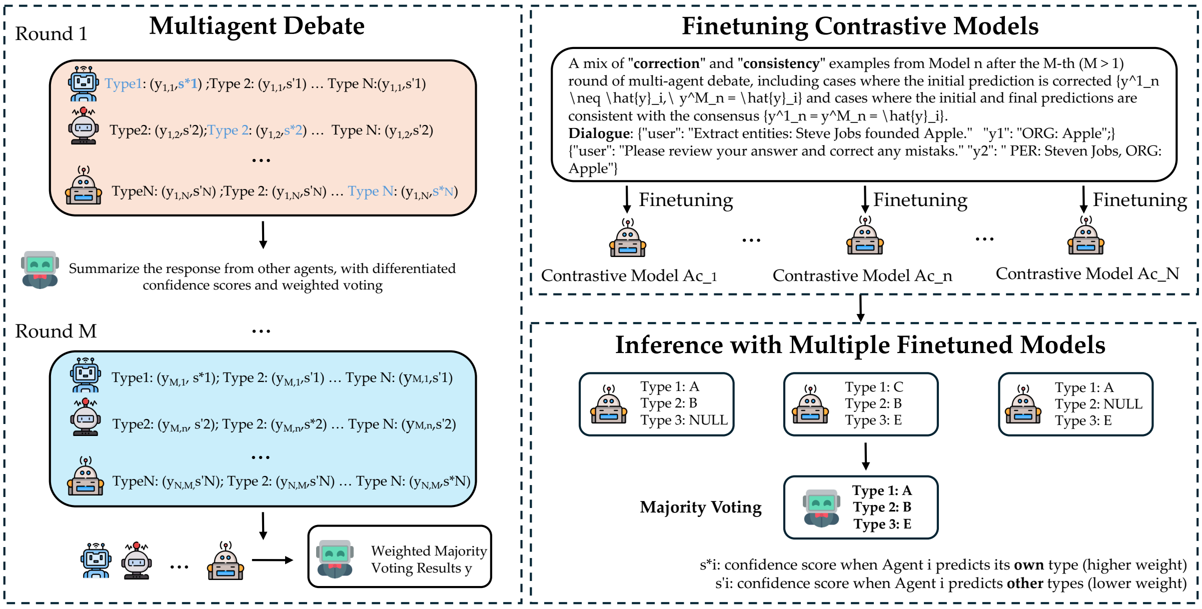


Figure 1: The overview of MAF-IE presents a multi-agent finetuning framework for zero-shot IE. We first employ type-specialized multi-agent debate and confidence-weighted voting to construct finetuning datasets. These datasets are then used to finetune the contrastive agents. We finetune contrastive models using reformatted dialogue-style data that includes final-round responses labeled by whether they match the weighted voting result, along with first-round responses from each type-specific agent to capture both "correction” and "consistency” signals, enabling the model to differentiate correct and incorrect predictions better. Finally, the finetuned models are combined via majority voting to produce more accurate predictions.

enabling each finetuned model to capture task-specific features while reducing the cost of multi-agent inference. Specifically, we propose a type-specified multi-agent collaboration system in which specialized agents engage in cross-type discussions to refine predictions and establish a feedback loop that improves extraction accuracy. Next, we leverage the outputs generated from these multi-agent interactions as pseudo-labeled data to finetune multiple LLMs, with each model trained on type-specific data to promote specialization across models. Finally, we combine the multiple finetuned models with a majority voting strategy at inference time to optimize the final predictions. Experimental results demonstrate that MAF-IE achieves significant improvements on seven IE datasets across six tasks in diverse domains under a zero-shot setting, spanning sentence- and document-level inputs as well as coarse- and fine-grained label schemas, validating the effectiveness and efficiency of our approach.

2 Related Works

LLMs for IE Recent advances in LLM-based IE have shown promise in tasks such as Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE). ChatIE (Wei et al., 2024) enhances IE through structured dialogue with ChatGPT, enabling interactive refinement. CODE4STRUCT (Wang et al., 2023b) and Code4UIE (Guo et al., 2023) formulate EE as a code generation problem, with the former representing event ontologies in code and the latter leveraging in-context learning with retrieved examples.

Multi-agent for IE The rise of LLM agents like GPTs (OpenAI, 2023a), LLaMAs (AI, 2024), and PaLM (Chowdhery et al., 2022) has enabled multi-agent collaboration through either cooperative (Zhang et al., 2024) or adversarial strategies (Aryan, 2024) to iteratively output refinement. DoA (Wang and Huang, 2024) introduces a debate optimization with few-shot learning for EE that iteratively refines outputs. EPASS (Hou et al., 2024) proposes a supervised dual-agent system for document-level RE, jointly modeling entity pairs and extracting cross-sentence evidence. TriageAgent (Lu et al., 2024) proposes a heterogeneous multi-agent clinical IE framework, where LLM agents collaborate via multi-round role-playing with confidence scoring and early stopping.

LLM Finetuning Several methods have been introduced for LLM finetunig, including single and multiple LLMs. RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2024) employ instruction tuning to improve the generated response to instructions. Supervised finetuning (SFT) (Pareja et al., 2024) employs large-batch and stacked training strategies on datasets to improve LLM generalization without relying on complex schedulers. GRPO (DeepSeek-AI et al., 2025) applies large-scale reinforcement learning directly on the base model, enabling the model to develop reasoning capabilities through self-evolution driven by reward signals autonomously. Multiagent finetuning (Subramaniam et al., 2025) introduces a self-improvement framework where LLM agents generate diverse reasoning data through multi-round debates to finetune multiple models, enabling performance improvements.

3 MAF-IE Framework

This section introduces MAF-IE, a **M**ulti-**A**gent **F**inetuning specifically designed for **I**nformation **E**xtraction. We first formalize the problem (Sec. 3.1), followed by a type-specialized multi-agent debate framework (Sec. 3.2). Next, we describe the construction of task-specific finetuning datasets (Sec. 3.3) and detail our multi-agent finetuning strategy, where each model is trained on data generated by all type-specific agents to achieve specialization (Sec. 3.4). Finally, we describe the inference process (Sec. 3.5). An overview of our approach can be seen in Figure 1.

3.1 Problem Definition

Given a natural language dataset $\mathcal{D}_{\text{task}} = \{x_i\}$, where each input x_i is a text sequence, the goal of IE is to produce structured outputs depending on the requirements. The NER identifies entity spans e in x_i as mentions and assigns each mention a type label $t \in \mathcal{T}$, where T is a predefined set of entity types (e.g.,*PER*, *ORG*, and *LOC*). The output is a set of labeled entity $\{(e, t) \mid e \in x_i, t \in \mathcal{T}\}$. Based on the identified entity set $E = \{e_1, e_2, \dots, e_k\}$, RE aims to detect and classify semantic relations $r_i \in \mathcal{R}$ between entities. The output is relation triples:$\{(e_p, r_i, e_q) \mid e_p, e_q \in E, r_i \in \mathcal{R}\}$. EE aims to detect event triggers $t \subseteq x_i$ in the text and classify their event types $e_t \in \mathcal{E}$, where $\mathcal{E}$ is a predefined event type (e.g., *Conflict:Attack*, *Life:Die*). For each identified event trigger, extract argument-role pairs $a_t = \{(r_j, e_j)\}$, where e_j represents entity mention and r_j is its semantic role in the event (e.g., *Agent*, *Victim*, *Time*). The output for x_i is structured event records $\mathcal{E}_i = \{(t, e_t, a_t) \mid t \subseteq x_i, e_t \in \mathcal{E}, a_t = \{(r_j, e_j)\}_{j=1}^{m}\}$. Fine-grained entity typing(FET) aims to assign fine-grained type labels to each marked entity mention $e_j \subseteq x_i$, where the type labels $\mathcal{T}_{\text{fine}}$ are drawn from a hierarchical type ontology (e.g., *Person/Artist/Actor*). The output for x_i is entity-type associations: $T_i = \{(e_j, S_j) \mid e_j \subseteq x_i, S_j \subseteq \mathcal{T}_{\text{fine}}\}$. See Appendix C.2 for more task definitions.

3.2 Multi-Agent Collaboration

We propose a type-specialized multi-agent collaboration framework to address key challenges in IE, including fine-grained type discrimination, boundary ambiguity, and complex semantic structures. The framework consists of N language models, instantiated as identical copies or finetuned variants of a shared base model, which engage in M debate rounds. Each agent specializes in a specific label type, generating predictions with higher confidence within its domain and providing auxiliary predictions for other labels to support cross-type verification. During each round r, agents exchange structured prompts containing their own and others' predictions, rationales, self-assessed confidence scores, and aggregated voting statistics. These confidence scores are recalibrated using a function $f(\cdot)$ like min-max normalization to ensure fair contribution weighting in the aggregation process. After M rounds, the final prediction is determined through confidence-weighted voting, formulated as: $\hat{y}^{(M)} = \arg\max_{y \in \mathcal{Y}} \sum_{n=1}^{N} f(p_n^{(M)}) \cdot \mathbf{1}(\hat{y}_n^{(M)} = y)$, where $\mathcal{Y}$ is the set of candidate entities, $p_n^{(M)}$ is agent A_n's original confidence score, $f(p_n^{(M)})$ is its calibrated value, and $\mathbf{1}(\hat{y}_n^{(M)} = y)$ indicates whether agent A_n voted for y. This voting strategy integrates consensus and agent confidence to improve the accuracy of type-specific extraction. We provide pseudocode in Algorithm 1 and 2.

3.3 Data Generation via Collaboration

We explore enhancing model performance by leveraging data generated through multi-agent debates among type-specialized agents. Specifically, we aim to construct diverse training datasets that capture label-specific knowledge and collaborative reasoning strategies. Given a set of natural language inputs $\mathcal{D}_{\text{task}} = \{x_i\}$, we apply the type-specialized multi-agent debate framework with N type agents and M debate rounds to generate structured responses for each input in $\mathcal{D}_{\text{task}}$. For each input x_i, the final prediction $\hat{y}_i$ is determined by weighted voting over the responses produced in the final round of debate. These predicted outputs are then used to construct a pseudo-labeled "ground truth" dataset $\{(xi, \hat{y}i)\}$. In the single-model finetuning setting, we subsequently train the model on all types of agents' generated responses y_i that match the final consensus prediction $\hat{y}_i$ for each x_i. While this approach is effective when the final predictions $\hat{y}_i$ are accurate, it often leads to stylistically homogeneous outputs with limited diversity. Consequently, repeatedly constructing datasets $x_i, \hat{y}_i$ for single model finetuning leads to diminishing returns, resulting in a performance plateau.

3.4 Finetuning Multiple Models

Our goal in multi-agent finetuning is to construct training datasets that promote both response diversity and high prediction accuracy for diverse IE tasks. To achieve this, we leverage data generated from multi-agent debates among type-specialized agents, capturing both label-specific knowledge and collaborative reasoning strategies. We provide pseudocode in Algorithm 3.

Finetuning Contrastive Models The role of a contrastive model is to improve decision accuracy by learning to distinguish correct from incorrect outputs through structured supervision. Contrastive agents A_n^C, which are constructed from base models, are trained on response trajectories collected from multi-agent debate outputs. For each input x_i, we collect the initial prediction y_n^1 and the final prediction y_n^M from each agent after M rounds of debate and compare them with the consensus output $\hat{y}_i$ through weighted voting.

To build a contrastive training dataset, we categorize the data into two types of samples based on the alignment between agent predictions and the consensus output. Correction samples capture cases where the initial prediction disagrees with the consensus, but the final prediction aligns with it, indicating successful error correction through debate: $\mathcal{D}_n^{C-} = \{(x_i, (y_n^1, \ldots, y_n^M)) \mid y_n^1 \neq \hat{y}_i,\ y_n^M = \hat{y}_i\}$. In contrast, consistency samples represent stable reasoning, where both the initial and final predictions agree with the consensus: $\mathcal{D}_n^{C+} = \{(x_i, (y_n^1, \ldots, y_n^M)) \mid y_n^1 = \hat{y}_i,\ y_n^M = \hat{y}_i\}$.

To facilitate contrastive learning, all training data are reformatted as multi-turn dialogues. Each dialogue starts with a task-specific prompt, followed by the agent's initial prediction y_n^1, a feedback prompt encouraging reflection on potential errors, and a revised prediction y_n^M aligned with the consensus $\hat{y}_i$. This dialogue structure extends beyond the traditional *question→answer* paradigm by incorporating a *feedback→correction* mechanism, enabling the model to learn both robust extraction and effective error-recovery strategies. To balance the influence of error correction and stable reasoning, we combine correction and consistency samples using a tunable weight w: $\mathcal{D}_n^C = w\mathcal{D}_n^{C-} + (1-w)\mathcal{D}_n^{C+}$. This process yields a set of contrastive datasets $\{\mathcal{D}_1^C, \ldots, \mathcal{D}_N^C\}$, which are used to fine-tune contrastive agents $\{\hat{A}_1^C, \ldots, \hat{A}_N^C\}$.

3.5 Inference

At inference time, we have a set of finetuned contrastive models that represent contrastive agents $\{\hat{A}_1^C, \ldots, \hat{A}_N^C\}$, each independently performing single-round inference for its designated task. The final output is determined through majority voting across all agent responses, which helps mitigate errors and improve overall performance on IE tasks.

Unlike reasoning tasks such as math or logical QA, multi-round debating among finetuned models degrades performance on structured IE tasks. Debating relies on generating diverse responses to expand the search space. In contrast, finetuning tends to converge model outputs, reducing response diversity and weakening debate effectiveness by producing more uniform and concentrated outputs. This convergence limits agent perspective diversity in multi-round debates. As a result, excessive debating leads to redundant refinements, added noise, and overall performance degradation. To mitigate this, we adopt a lightweight voting strategy where task-specialized models generate independent predictions in parallel, and majority voting aggregates these outputs to achieve consistency and efficiency. We provide pseudocode in Algorithm 4.

4 Experiments

We evaluate MAF-IE on a diverse set of IE tasks using strict span-level matching and report micro-F1 scores against GPT-3.5 zero-shot baselines. We further assess generalization on GPT-4 and clinical tasks. See Appendix A.1 for details.

4.1 Experimental Setup

We propose a novel multi-agent finetuning framework for zero-shot IE, evaluated against single-model and multi-agent framework baselines.

Tasks and Datasets For a comprehensive evaluation, we examine MAF-IE on seven datasets for five IE tasks: (1) for named entity recognition (NER): (i) **CoNLL04** (Carreras and Màrquez, 2004), (ii) **BC5CDR** (Li et al., 2016) ; (2) for relation extraction (RE): (i) **CoNLL04** (Carreras and Màrquez, 2004) (ii) **NYT** (Zeng et al., 2018); (3) for event extraction(EE): (i)**ACE05-E** [1] (ii) **MACCROBAT-EE** (Ma et al., 2023); (4) for fine-grained entity typing (FET): (i) **OntoNotes** (Gillick et al., 2016); (5) for document-level RE: (i) **DocRed** (Yao et al., 2019). Please refer to Appendix A for more information about tasks and datasets.

Baselines We conduct our main experiments using both GPT-3.5 and GPT-4. We employ the (1) Type-Agents, where each agent specializes in a specific label type without inter-agent interaction and (2) Multiagent finetuning (MAFT) (Subramaniam et al., 2025), which employs general-built LLMs in iterative collaborative reasoning as the baselines for all zero-shot IE tasks.

NER and RE We consider Direct prompting a fundamental single-model baseline for both tasks. This method jointly identifies and organizes outputs in a one-step prompt. For NER, we additionally include: (3) Self-consistency (Wang et al., 2023c), which aggregates multiple outputs via voting to improve stability; (4) Soft Self-consistency (Wang et al., 2024a), which softens voting decisions using uncertainty-aware aggregation. For RE, we further compare: (5) G&O (Li et al., 2024b), a pipeline-based approach that generates triplets and then organizes them into structured outputs.

EE We compare MAF-IE against the following additional baselines: (3) ChatGPT-14 (Li et al., 2023a), the first study evaluating ChatGPT's zero-shot performance on IE tasks. (4) ChatIE (Wei et al., 2024), a multi-turn QA framework that first identifies all event types, then performs IE for each identified type. (5) G-PTLM (Lin et al., 2023), a prompting-based model that encodes argument constraints to regularize event argument predictions, and (6) CODE4STRUCT (Wang et al., 2023b) formulates EE as a code generation problem, and represents event ontology in Python code expression.

Fine-grained Entity Typing We compare MAF-IE against additional baselines on the FET task, including (3) ONTOTYPE, combining BERT-based prompting with RoBERTa-MNLI entailment for ontology-aware selection; (4) ZOE (Zhou et al., 2018), which aligns entities to Wikipedia entries via Boolean functions over Freebase types; (5) DZET (Obeidat et al., 2019), which uses distributed description representations for semantic alignment.

[1] `https://catalog.ldc.upenn.edu/LDC2006T06`

Tasks (→) / **Baselines** (↓) / **Metrics** (→)	**NER** F1-Score	**RE** F1-Score
Single model		
GPT-3.5 (OpenAI, 2023a)	58.15	<u>34.72</u>†
+ G&O (Li et al., 2024b)	-	33.50
+ Self-consistency (Wang et al., 2023c)	60.48	-
+ Soft Self-consistency (Wang et al., 2023c)	55.13	-
Multi-agent framework		
+ MAFT (Subramaniam et al., 2025)	<u>61.12</u>†	20.51
+ **MAF-IE** (Type-agent w/o debate)	55.73	29.97
+ **MAF-IE** (Multi-agent Collaboration)	**66.83**	**36.47**
Fine-tune (FT)		
+ MAFT (Subramaniam et al., 2025)	61.12	20.51
+ **MAF-IE** (Single FT)	64.21	28.63
+ **MAF-IE** (Multiple FT)	62.51	33.47
GPT-4 (OpenAI, 2024b)	66.59	21.01
+ **MAF-IE** (Multi-agent Collaboration)	71.46	44.03
+ **MAF-IE** (Single FT)	67.26	38.26
+ **MAF-IE** (Multiple FT)	63.65	41.76

Table 1: Main results on CONLL04 for NER and RE tasks in zero-shot setting. Bold indicates the best performance.† marks the second-best. Notations are consistent across tables.

Implementation Details The proposed system is flexible, allowing any LLM to serve in any arbitrary agent role defined within the framework. We conduct zero-shot experiments using GPT-3.5-Trubo (OpenAI, 2023b) and GPT-4 (OpenAI, 2024b). We set the number of collaboration iterations to 2 and perform single-step inference. We set the temperature to 1 to ensure reproducibility. Please refer to Appendix B for more details.

5 Main Results

MAF-IE outperforms zero-shot baselines for NER and RE tasks Table 1 shows that MAF-IE consistently outperforms all zero-shot baselines on CONLL04 NER and RE with GPT-3.5 and GPT-4. With GPT-3.5, MAF-IE achieves gains of 5.71% (NER) and 15.96% (RE) over the multi-agent baseline, and 1.75% over G&O on RE. Finetuning further improves performance by 5.67% (NER) and 10.78% (RE), while our single finetuned model surpasses direct prompting by 3.73% on NER. Applying MAF-IE to GPT-4 achieves the best results on both tasks, with 71.46% (NER) and 44.03%

Tasks (→) Baselines (↓) / Metrics (→)	ED F1	EAE F1	EE F1
Single model			
GPT-3.5 (OpenAI, 2023a)			
+ ChatGPT-14 (Li et al., 2023a)	17.1	28.9	16.6
+ ChatIE (Wei et al., 2024)	-	29.5	-
+ G-PTLM (Lin et al., 2023)	-	31.2	-
CODE4STRUCT (Wang et al., 2023b)	-	37.8	-
Multi-agent framework			
+ MAFT (Subramaniam et al., 2025)	23.93	21.73	18.05
+ **MAF-IE** (Type-agent w/o debate)	23.85	35.98	16.57
+ **MAF-IE** (Multi-agent Collaboration)	36.98†	**38.87**	**34.32**
Fine-tune (FT)			
+ MAFT (Subramaniam et al., 2025)	21.36	17.16	14.94
+ **MAF-IE** (Single FT)	36.21	34.97	22.41
+ **MAF-IE** (Multiple FT)	**41.32**	36.41†	24.98†
GPT-4 (OpenAI, 2024b)			
+ **MAF-IE** (Multi-agent Collaboration)	54.01	49.43	45.46
+ **MAF-IE** (Multiple FT)	43.18	41.56	35.38

Table 2: Main results on ACE05 for ED, EAE, and EE tasks in zero-shot setting.

Metrics (→) Baselines (↓)	Accuracy (%)	F1 (%)
Single model		
GPT-3.5 (OpenAI, 2023a)		
Distant Supervision via KBs		
+ DZET(Obeidat et al., 2019)	23.1	28.1
+ ZOE (Zhou et al., 2018)	50.7	60.8
Transfer Learning		
+ OTyper (Yuan and Downey, 2018)	31.8	36.0
+ MZET (Zhang et al., 2020)	33.7	43.7
Annotation-Free		
+ ChatGPT-14 (Li et al., 2023a)	-	73.4
+ OntoType (Li et al., 2023a)		
- ChatGPT Prompt 1	27.7	37.5
- ChatGPT Prompt 2	31.3	41.3
- ChatGPT Prompt 3	24.7	33.8
- Original Ontology	65.7†	73.4†
Multi-agent framework		
+ MAFT (Subramaniam et al., 2025)	46.81	53.61
+ **MAF-IE** (Type-agent w/o debate)	11.05	18.91
+ **MAF-IE** (Multi-agent Collaboration)	**66.91**	**74.51**
GPT-4 (OpenAI, 2024b)		
+ **MAF-IE** (Multi-agent Collaboration)	76.14	84.85

Table 3: Main results on OntoNotes for Fine-grained entity typing task in zero-shot setting.

(RE), demonstrating its scalability across models.

MAF-IE outperforms zero-shot baselines for EE tasks Table 2 shows that MAF-IE achieves strong zero-shot F1 improvements on ACE05 with GPT-3.5, outperforming the multi-agent baseline by 13.05% (ED), 17.14% (EAE), and 16.27% (EE). Compared to ChatGPT-14, MAF-IE achieves gains of 19.88% (ED), 9.97% (EAE), and 17.72% (EE), and exceeds the second-best EAE baseline, CODE4STRUCT, by 1.07%. In the finetuning setting, MAF-IE achieves even larger improvements, with gains of 19.96% (ED), 19.45% (EAE), and 9.54% (EE), and further improves debating accuracy by 4.34% on ED. Finally, our finetuned single model surpasses the best single-model baseline by 19.11% (ED), 5.81% (EE), and achieves an average 5.10% improvement on EAE across all zero-shot single LLM baselines.

MAF-IE outperforms zero-shot baselines for Fine-grained entity typing Table 3 shows that MAF-IE achieves the best zero-shot F1 on OntoNotes with GPT-3.5, consistently outperforming ChatGPT-14, ZOE, and direct prompt methods (Komarlu et al., 2024a), with gains of 1.11%, 13.71%, and 33.21%, respectively. MAF-IE also surpasses the state-of-the-art OntoType by 1.11%.

MAF-IE generalizes across diverse IE settings (long-document RE) and domains (news, biomedicine) We further validate the generalizability of our multi-agent collaboration framework across diverse IE tasks and domains, including document-level RE (DocRed), biomedical NER (BC5CDR), clinical EE (MACCROBAT), and RE on the NYT dataset. Please see Appendix C.

6 Ablation Studies

In this section, we investigate the effectiveness of MAF-IE, its impact on enhancing response diversity, and its ability to generalize to unseen datasets in a zero-shot setting.

Multi-agent debate with different number of rounds We evaluate MAF-IE on the CoNLL04 NER task using GPT-3.5 with varying numbers of debate rounds, and compare it against prior work MAFT, as illustrated in Figure 2(a). We observe that increasing the number of rounds beyond two leads to diminishing returns, with both methods reaching a performance plateau. Excessive debate rounds provide limited gains for IE tasks, as early rounds already capture most correct entities, while further iterations risk over-refinement and noise accumulation. We notice recent work MoA (Wang et al., 2024b) uses multiple heterogeneous LLMs to exploit complementary strengths, while MAF-IE focuses on improving a single base model through

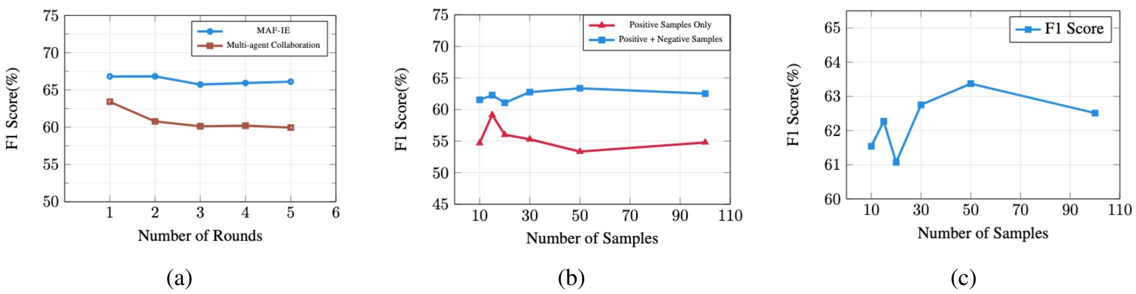


Figure 2: Ablation study results. (a) shows F1(%) across different debate rounds; (b) shows F1(%) with varying numbers of training samples; (c) shows the impact of contrastive datasets. All results are evaluated on the CoNLL04 NER task using GPT-3.5.

multi-agent finetuning, offering a more economical and lightweight solution. Extending our framework to heterogeneous agents is left for future work.

Multi-agent FT with different dataset construction strategies We compare two data construction strategies based on multi-agent debate. The first uses only consensus outputs as positive examples, while the second builds a contrastive dataset with both positives (aligned initial responses) and negatives, where negatives are represented as dialogue trajectories (question $\rightarrow$ incorrect answer $\rightarrow$ feedback $\rightarrow$ revised answer). As shown in Figure 2(b), the contrastive strategy improves average F1 by 6%, increasing true positives and reducing false negatives compared to the positive-only.

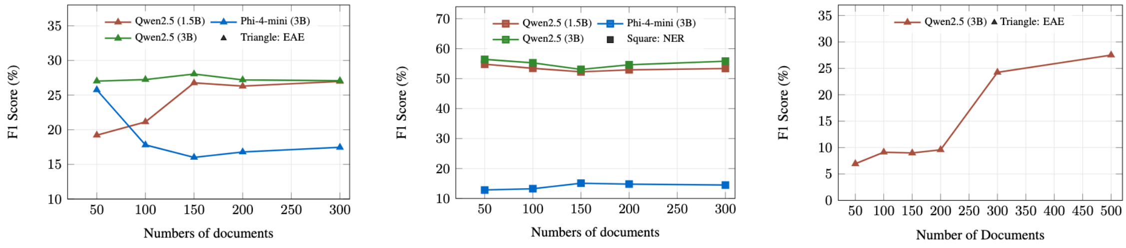
Multi-agent FT with different data selection strategies We investigate how the strategy for training data selection impacts multi-agent finetuning, random sampling, and confidence-based selection guided by scores assigned by a GPT-3.5 judge on the CONLL04 NER task. As shown in Table 5, the confidence-based strategy achieves a higher average F1 (62.72% vs. 61.94%) and lower variance (0.22 vs. 1.56) with 50 samples, demonstrating more stable and reliable performance in low-resource settings.

Multi-agent FT with different numbers of examples We investigate how the number of examples from the training data affects the performance of multi-agent finetuning on the CONLL04 NER task with GPT-3.5. As shown in Figure 2(c), the F1 score does not consistently improve as the training examples increase. Our results indicate that finetuning each model with 15-20 examples per type label yields optimal performance, likely due to a balance between sufficient task coverage and the overfitting risk.

Final answer generation from multiple finetuned models As shown in Table 8 in Appendix E, the majority voting improves the overall F1 score of individual models. It achieves the highest recall, demonstrating its effectiveness in enhancing robustness and reducing false negatives without sacrificing precision.

Finetune small language models We evaluate the performance of finetuning Qwen2.5 (1.5B), Qwen2.5 (3B), and Phi-4-mini (3B) on generated data for NER (CoNLL04) and EAE (ACE05), comparing supervised finetuning (SFT) and GRPO (Mroueh, 2025). As shown in Figures 3(a) and (b), SFT results indicate that Qwen2.5 (3B) consistently achieves the best and most stable performance, peaking with around 200 training examples. Qwen2.5 (1.5B) achieves moderate improvements, while Phi-4-mini performs poorly on both tasks, showing low and stagnant F1 scores, suggesting limited capabilities to benefit from training data. Figure 3(c) shows the GRPO performance on Qwen2.5-(3B) for the EAE task, indicating strong data dependence, with performance steadily improving as the amount of training data increases. Interestingly, a simple reward design based on output format and accuracy proves more effective than complex alternatives. However, GRPO comes with significant time costs, requiring over 7 hours for 500 examples and several days to reach GPT-3.5-level performance with thousands of examples. Moreover, the high cost of large-scale annotations further limits its scalability in low-resource and real-world applications.

Compared with the few-shot setting We evaluate few-shot prompting on the CoNLL04 NER task using GPT-3.5. As shown in Table 4, adding more in-context examples provides only marginal improvements, with performance quickly plateau-



(a) (b) (c)

Figure 3: Ablation study results. (a) shows EAE F1(%) across different small language models on SFT; (b) shows NER F1(%) across different small language models on SFT; (c) shows EAE performance of Qwen2.5(3B) with GRPO across different number of samples

Few-shot Method	Precision	Recall	F1
5-shots	56.54	81.05	66.61
10-shots	57.73	82.06	67.78
15-shots	57.24	82.06	67.44

Table 4: The results of few-shot learning on CONLL04 NER task with GPT-3.5.

ing. This highlights the limited generalization ability of LLMs when relying on static examples for IE tasks. Although few-shot prompting appears cost-effective, its actual gains are minimal, and approaches that depend on carefully designed examples often require complex designs and costly training, limiting practical utility. In contrast, our multi-agent finetuning framework provides a more practical and scalable solution. It requires only a one-time collaboration and finetuning process, after which the resulting models can be directly applied to unseen datasets without further adaptation, achieving both cost and time efficiency for real-world IE deployment.

Efficiency & scalability study We evaluate the cost and time per data point on CONLL04 NER and RE tasks with GPT-3.5. As shown in Appendix H.1 Table 23 and H.2 Table 24, we observe that finetuned parallel inference reduces latency by 42% on NER and 50% on RE, matching single-agent speed while avoiding the overhead of multi-round debate. Compared to multi-agent debate, finetuned parallel inference improves cost efficiency by 90% on NER and 84% on RE, offering a practical and scalable alternative that retains most of the performance benefits while significantly reducing costs. More analysis is provided in Appendix H.

Case Study & Error Analysis We compare MAF-IE with MAFT on the CONLL04 NER with GPT-3.5, conducting a comprehensive error analysis covering overall and type-specific improvements, representative case studies, and the incremental impact of each debate round. Specifically, Table 6 summarizes overall and entity-level gains, Table 25 presents case studies of error corrections, and Table 7 quantifies stepwise improvements across debate rounds. To better understand the source of these improvements, we further analyze how MAF-IE addresses key challenges in structured IE. It improves type discrimination through agent specialization, mitigates boundary ambiguity via cross-type verification, and enhances robustness on complex semantics by aggregating diverse rationales through cross-agent voting. *All tables mentioned above and additional details are provided in Appendix I.*

7 Conclusion

In this paper, we have introduced MAF-IE, a novel multi-agent finetuning framework that improves the efficiency and effectiveness of LLMs for zero-shot IE. By leveraging a society of specialized agents that collaboratively solve IE tasks through multi-agent debate and confidence-weighted voting, MAF-IE addresses key limitations of single LLMs on IE. This system enables the distillation of collaborative knowledge into a set of finetuned models, achieving substantial performance gains across a broad range of structured IE tasks. Importantly, MAF-IE is generalizable and scalable to both open-source and proprietary language models and provides a more efficient alternative to costly multi-agent inference. Additionally, MAF-IE can be combined with other advanced finetuning paradigms such as GRPO and extended to heterogeneous model agents, which we leave for future work. This work sets up the foundation for advancing efficient and scalable zero-shot IE with LLMs.

Limitations

In contrast to existing approaches that rely on direct inference or finetuning of a single model, multi-agent finetuning introduces computational overhead during both training and inference, as it requires maintaining and running multiple model instances. Specifically, we identify the following limitations of MAF-IE:

Scalability in Multi-Agent Collaboration As the number of agents increases, coordination complexity grows. Managing conflicts and ensuring convergence in large-scale settings require further optimization to prevent excessive inference time.

Dependency on Model Accuracy The framework relies on LLMs' reasoning capabilities, which can still produce hallucinated or inconsistent outputs. Additionally, due to the risk posed by the inherent instability of large language model generation, biases, trust issues, or other uncertainties may arise, potentially undermining the reliability of the extracted information.

Ontology Constraints Our approach operates within predefined entity and relation ontologies, limiting adaptability to open-domain or evolving schemas. Extending it to dynamic ontologies would require additional mechanisms for expansion and adaptation.

Ethical Statement

In this work, we propose a multi-agent finetuning method to improve LLM performance on the important and fundamental task of information extraction. We do not anticipate any ethical issues regarding the topics of this research.

References

Meta AI. 2024. Llama 3.1 - 70b. https://huggingface.co/meta-llama/Llama-3.1-70B. Accessed: May 2, 2025.

Alpindale. 2024. Wizardlm-2-8x22b. https://huggingface.co/alpindale/WizardLM-2-8x22B. Accessed: May 2, 2025.

Prakash Aryan. 2024. Llms as debate partners: Utilizing genetic algorithms and adversarial search for adaptive arguments. *Preprint*, arXiv:2412.06229.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *Preprint*, arXiv:2308.07201.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *Preprint*, arXiv:2309.13007.

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, and 48 others. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2016. Context-dependent fine-grained entity type tagging. *Preprint*, arXiv:1412.1820.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2023. Retrieval-augmented code generation for universal information extraction. *Preprint*, arXiv:2311.02962.

Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang. 2024. ProgGen: Generating named entity recognition datasets step-by-step with self-reflexive large language models. In *Findings of the Association for*

Computational Linguistics: ACL 2024, pages 15992–16030, Bangkok, Thailand. Association for Computational Linguistics.

Wenlong Hou, Ning Jia, Xianhui Liu, Weidong Zhao, and Zekai Wang. 2024. A multiagent-based document-level relation extraction system with entity pair awareness and sentence significance. *IEEE Systems Journal*, 18(4):1905–1916.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Tanay Komarlu, Minhao Jiang, Xuan Wang, and Jiawei Han. 2024a. Ontotype: Ontology-guided and pre-trained language model assisted fine-grained entity typing. *Preprint*, arXiv:2305.12307.

Tanay Komarlu, Minhao Jiang, Xuan Wang, and Jiawei Han. 2024b. Ontotype: Ontology-guided and pre-trained language model assisted fine-grained entity typing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 1407âĂŞ1417, New York, NY, USA. Association for Computing Machinery.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *Preprint*, arXiv:2304.11633.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database (Oxford)*, 2016:baw068.

Junpeng Li, Zixia Jia, and Zilong Zheng. 2023b. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5495–5505, Singapore. Association for Computational Linguistics.

Xingzuo Li, Kehai Chen, Yunfei Long, and Min Zhang. 2024a. Llm with relation classifier for document-level relation extraction. *Preprint*, arXiv:2408.13889.

Yinghao Li, Colin Lockard, Prashant Shiralkar, and Chao Zhang. 2023c. Extracting shopping interest-related product types from the web. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7509–7525, Toronto, Canada. Association for Computational Linguistics.

Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024b. A simple but effective approach to improve structured language model output for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5133–5148, Miami, Florida, USA. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Zizheng Lin, Hongming Zhang, and Yangqiu Song. 2023. Global constraints with prompting for zero-shot event argument classification. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2527–2538, Dubrovnik, Croatia. Association for Computational Linguistics.

Keming Lu, I-Hung Hsu, Wenxuan Zhou, Mingyu Derek Ma, and Muhao Chen. 2022. Summarization as indirect supervision for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6575–6594, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Meng Lu, Brandon Ho, Dennis Ren, and Xuan Wang. 2024. TriageAgent: Towards better multi-agents collaborations for large language model-based clinical triage. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5747–5764, Miami, Florida, USA. Association for Computational Linguistics.

Mingyu Derek Ma, Alexander Taylor, Wei Wang, and Nanyun Peng. 2023. DICE: Data-efficient clinical event extraction with generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15898–15917, Toronto, Canada. Association for Computational Linguistics.

Microsoft. 2024. Phi-4-mini-instruct. `https://huggingface.co/microsoft/Phi-4-mini-instruct`. Accessed: May 2, 2025.

Youssef Mroueh. 2025. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *Preprint*, arXiv:2503.06639.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 807–814, Minneapolis, Minnesota. Association for Computational Linguistics.

OpenAI. 2023a. Chatgpt: Openai's language model. Accessed: November 10, 2023.

OpenAI. 2023b. Gpt-3: Openai's language model. https://www.openai.com/. Accessed: November 10, 2023.

OpenAI. 2024a. Introducing gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: April 27, 2025.

OpenAI. 2024b. Introducing gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: May 2, 2025.

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

Aldo Pareja, Nikhil Shivakumar Nayak, Hao Wang, Krishnateja Killamsetty, Shivchander Sudalairaj, Wenlong Zhao, Seungwook Han, Abhishek Bhandwaldar, Guangxuan Xu, Kai Xu, Ligong Han, Luke Inglis, and Akash Srivastava. 2024. Unveiling the secret recipe: A guide for supervised fine-tuning small llms. *Preprint*, arXiv:2412.13337.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A. Plummer, Zhaoran Wang, and Hongxia Yang. 2024. Let models speak ciphers: Multiagent debate through embeddings. *Preprint*, arXiv:2310.06272.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Simonycl. 2024. Qwen 2.5 - 70b instruct. https://huggingface.co/simonycl/qwen-2.5-qwen-2.5-70b-instruct. Accessed: May 2, 2025.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. 2025. Multiagent finetuning: Self improvement with diverse reasoning chains. *Preprint*, arXiv:2501.05707.

Lijun Sun, Yijun Yang, Qiqi Duan, Yuhui Shi, Chao Lyu, Yu-Cheng Chang, Chin-Teng Lin, and Yang Shen. 2025. Multi-agent coordination across diverse applications: A survey. *Preprint*, arXiv:2502.14743.

Qwen Team. 2024a. Qwen1.5-110b-chat. https://huggingface.co/Qwen/Qwen1.5-110B-Chat. Accessed: May 2, 2025.

Qwen Team. 2024b. Qwen2.5-1.5b-instruct. https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct. Accessed: May 2, 2025.

Qwen Team. 2024c. Qwen2.5-3b-instruct. https://huggingface.co/Qwen/Qwen2.5-3B-Instruct. Accessed: May 2, 2025.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024a. Soft self-consistency improves language model agents. *Preprint*, arXiv:2402.13212.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024b. Mixture-of-agents enhances large language model capabilities. *Preprint*, arXiv:2406.04692.

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

Sijia Wang and Lifu Huang. 2024. Debate as optimization: Adaptive conformal prediction and diverse retrieval for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16422–16435, Miami, Florida, USA. Association for Computational Linguistics.

Xingyao Wang, Sha Li, and Heng Ji. 2023b. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

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. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

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

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *Preprint*, arXiv:2402.14568.

Zheng Yuan and Doug Downey. 2018. Otyper: a neural architecture for open named entity typing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Tao Zhang, Congying Xia, Chun-Ta Lu, and Philip Yu. 2020. MZET: Memory augmented zero-shot fine-grained named entity typing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 77–87, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yiqun Zhang, Xiaocui Yang, Shi Feng, Daling Wang, Yifei Zhang, and Kaisong Song. 2024. Can llms beat humans in debating? a dynamic multi-agent framework for competitive debate. *Preprint*, arXiv:2408.04472.

Wanjia Zhao, Mert Yuksekgonul, Shirley Wu, and James Zou. 2025. Sirius: Self-improving multi-agent systems via bootstrapped reasoning. *Preprint*, arXiv:2502.04780.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-shot open entity typing as type-compatible grounding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2065–2076, Brussels, Belgium. Association for Computational Linguistics.

Weiran Zhu, Xinzhi Wang, Xue Chen, and Xiangfeng Luo. 2024. Refining chatgpt for document-level relation extraction: A multi-dimensional prompting approach. In *Advanced Intelligent Computing Technology and Applications: 20th International Conference, ICIC 2024, Tianjin, China, August 5âĂŞ8, 2024, Proceedings, Part III*, page 190âĂŞ201, Berlin, Heidelberg. Springer-Verlag.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Dataset Details

We evaluate MAF-IE on seven diverse IE datasets, including CONLL04, NYT (Zeng et al., 2018), BC5CDR (Li et al., 2016), OntoNotes (Gillick et al., 2016), DocRed (Yao et al., 2019), ACE05, and MACCROBAT, which covers NER, RE, EE, and fine-grained entity typing tasks across both sentence-level and document-level inputs, and spanning coarse- and fine-grained settings. All results are reported under strict span-level full-matching criteria, where only predictions that perfectly match the ground-truth entity spans and labels are counted as true positives. We use GPT-3.5-turbo and report micro-averaged F1 scores for fair comparison with existing zero-shot baselines. Additionally, to assess the generalization capability of our framework, we further evaluate it with GPT-4 and extend the evaluation to the clinical datasets. The dataset statistics for all evaluation benchmarks are summarized as follows: CoNLL04 in Table 18, NYT-RE in Table 22, BC5CDR-NER in Table 21, OntoNotes in Table 19, DocRED in Table 17, ACE05 in Table 20, and MACCROBAT-EE in Table 16. The proposed data construction procedure for contrastive model finetuning is detailed in Algorithm 3.

A.1 Metrics and Evaluation

We compute micro-averaged precision, recall, and F1-score [2] using a strict span-level matching.

For NER and RE tasks, we conduct experiments on the CoNLL04 test dataset (Carreras and Màrquez, 2004), including three entities and five relation types. We additionally conduct NER on the BC5CDR test dataset and RE on the NYT test dataset.

For the EE task, we evaluate on two public event extraction test datasets: ACE05-E [3] and MACCROBAT-EE (Ma et al., 2023). Following prior split work(Lin et al., 2020), we evaluate three subtasks: (i) Event Detection (ED), where event types are given and the goal is to identify triggers; (ii) Event Argument Extraction (EAE), where both event types and triggers are provided; and (iii) Joint EE. We report Exact Match F1 for ED and Argument Head F1 for EAE and EE.

For fine-grained entity typing, we evaluate the performance on Ontonotes (Gillick et al., 2016). The basic statistics of the dataset are shown in

[2] https://scikit-learn.org/stable/index.html
[3] https://catalog.ldc.upenn.edu/LDC2006T06

Appendix 19. We followed previous work (Komarlu et al., 2024b) that each entity mentioned is labeled with a fine-grained label represented as a path within the ontology. The ontologies have a maximum depth of three and contain four high-level types (e.g., LOC, PER, and ORG).

Details During pre-processing for the NER task, we extract entities for each ontology-defined type from every document, constructing type-specific ground truth annotations. If a document lacks entities of a given type, the corresponding list remains empty. For RE, we extract head-tail entity pairs for each relation type, leaving the output empty when no valid pairs exist.

During post-processing, LLMs often introduce noise due to their generative nature, leading to discrepancies between outputs and the original text. Common issues include extraneous content, spacing inconsistencies, tense variations, and redundant acronym clarifications. These inconsistencies are particularly prevalent in large models, which may alter phrasing or terminology when extracting entities or relationships.

To mitigate these issues, we filter noisy content by matching generated outputs with original sentences. For RE, we format the output as [head: head_entity, tail: tail_entity] and validate entity pairs for each relation type. Consequently, we obtain structured entity lists: in NER, entities of a specific type per document; in RE, head-tail entity pairs per relation type.

To maintain the correct logical order between the head entity and tail entity, we provide natural language explanations that explicitly define the expected entity types for each relation. This ensures that extracted entities align with their intended semantic roles and follow the correct relationship direction. By clarifying entity-role expectations, we aim to mitigate errors such as entity misidentification or head-tail position errors caused by position bias or incorrect ordering. Furthermore, enforcing role consistency through relation constraints reduces relational confusion, enhancing extraction accuracy.

We follow the traditional pipeline for fine-tuning inference on a single GPT model, sequentially processing each sentence for NER and RE across all labels. Finally, we evaluate model performance using precision, recall, and F1-score, measuring alignment between predicted and ground truth entity spans. We use a full match criterion, requiring exact span agreement between predictions and ground truth to maintain consistency with traditional methods. For instance, in the sentence from doc_id 3: "He's working for the White House", the ground truth entity labeled as ORG_Agent might be:

```
doc_id 3: [White House]
```

If the ORG_agent predicts:

```
doc_id 3: [the White House]
```

with the additional word "the" in the span, it would be counted as both a false positive and a false negative under the full match evaluation. Similarly, if the ORG_Agent label incorrectly includes "White House" in its list, it would also be considered incorrect under the matching criteria. This rigorous evaluation method ensures a thorough assessment of the model's performance by capturing subtle span mismatches that could impact entity recognition accuracy.

A.2 Document-level Relation Extraction

We apply MAF-IE on document-level RE task on DocRed (Yao et al., 2019), which deeper verify the effectiveness of our method.

Problem definition Given a document D that includes a set of sentences $X_D = \{x_i\}_{i=1}^{k}$ and a set of entities $E_D = \{e_i\}_{i=1}^{n}$, document-level relation extraction aims to predict a subset of relations from $R \cup \{\text{NA}\}$ for all entity pairs (e_s, e_o) where $s, o = 1, \ldots, n$ and $s \neq o$. Here, R represents a predefined set of relation types, e_s and e_o denote the subject and object entities respectively, and NA indicates no relation between the entities. An entity e_i can appear multiple times within a document through its mentions $M_i = \{m_j^i\}_{j=1}^{N_i}$, where m_j^i represents the j-th mention of e_i, and N_i is the number of mentions. During test time, the model is required to predict relation labels for all possible entity pairs in the document. Table 17 presents the statistics of DocRed.

A.3 Clinical Event Extraction

We apply MAF-IE on MACCROBAT-EE, a clinical EE dataset that consists of 200 pairs of English clinical case reports from PubMed, accompanying annotation files with partial event annotation provided by 6 annotators with prior experience in biomedical annotations. Table 16 presents the statistics of MACCROBAT-EE.

Selection Method	Time-1	Time-2	Time-3
Randomly selection			
10-data points	65.58	64.59	62.19
15-data points	64.38	63.87	61.71
20-data points	61.17	59.09	62.42
30-data points	58.45	57.94	58.37
50-data points	63.54	61.79	60.49
100-data points	63.07	62.27	61.36
Confidence-score selection			
10-data points	61.54	61.79	60.21
15-data points	62.51	62.37	61.17
20-data points	61.07	62.97	62.32
30-data points	62.75	61.81	64.57
50-data points	63.37	62.28	62.51
100-data points	62.51	62.35	61.51

Table 5: F1 scores (%) (mean) of different examples selection strategies.

B Implementation Details

The proposed system is flexible, allowing any LLM to serve in any arbitrary agent role defined within the framework. We conduct zero-shot experiments using GPT-3.5-turbo (OpenAI, 2023b) and GPT-4 (OpenAI, 2024b). Each label is assigned a dedicated type agent, forming a one-to-one mapping with the label set. We set the number of collaboration iterations to 2 and perform single-step inference. We set the number of finetuned models to 3 for all tasks. The judge agent to select data points is powered by GPT-3.5-turbo. We set the temperature to 1 to ensure reproducibility. For supervised finetuning and reinforcement learning fine-tuning baselines, we use Qwen2.5-1.5B (Team, 2024b), Qwen2.5-3B (Team, 2024c), and Phi4-mini-3B (Microsoft, 2024). All fine-tuning experiments are conducted on NVIDIA A100 GPUs. Our reinforcement learning and related experiments on open-source models were conducted on clusters with four H100 or A100 GPUs, with each model consuming 80GB to 160GB of memory and requiring 24 to 48 hours of multi-GPU inference.

C Additional Experimental Results

C.1 Fine-grained Entity Typing

We conduct experiments on the test set of the OntoNotes dataset (Komarlu et al., 2024b), assigning each type label from different levels of the ontology to a dedicated agent to evaluate the effectiveness of our multi-agent framework on large-scale, fine-grained classification tasks. The OntoNotes dataset contains a total of 89 type labels, and we deploy 89 specialized agents accordingly to perform this task in a distributed and parallel manner. Table 3 shows our results on the test set. MAF-IE achieves the best zero-shot performance on this dataset. Compared to the state-of-the-art zero-shot fine-grained entity typing methods, ChatGPT-14 and ZOE, MAF-IE achieves absolute F1 improvements of 3.71% and 16.31%, respectively. Moreover, compared to direct prompt methods (Komarlu et al., 2024a) with GPT-3.5, MAF-IE achieves a substantial F1 improvement of 35.81%. MAF-IE also surpasses the previous state-of-the-art method, OntoType, by 3.71% in F1 score.

Multiagent Collaboration Framework for Entity Typing on the OntoNotes Dataset To address the challenge of fine-grained entity typing, we design a multi-agent collaboration framework based on type-agent collaboration and multi-round debate, tailored explicitly to the hierarchical entity type schema of the OntoNotes dataset. This framework constructs a multi-level entity typing system through three key stages: type-specialized agent modeling, multi-round interactive debate, and hierarchical weighted decision-making. We begin by analyzing the entity type hierarchy in the OntoNotes dataset and constructing a three-level hierarchical structure, ranging from coarse-grained to fine-grained types. This structure includes main categories (PER, LOC, ORG, OTHER), subcategories, and finer-grained subtypes. For each type, the system instantiates a specialized agent with expert knowledge specific to that type, making it an expert in its domain. When processing a new entity, the system initiates a multi-stage collaboration process. In the first stage, all agents independently analyze the entity's contextual and semantic features to form preliminary judgments. In the subsequent debate stage, agents exchange their perspectives, present supporting or opposing arguments, and dynamically refine their decisions based on the insights shared during the debate. After the debate concludes, the system applies a hierarchical, weighted voting mechanism to aggregate the opinions of all agents. In this process, specialized experts are assigned higher voting weights. The voting follows a hierarchical decision principle, prioritizing consensus at the most fine-grained level and falling back to higher-level categories if no consensus is reached. This framework effectively simulates collaborative decision-making among human experts, enabling the system to handle the complexity and uncertainty of entity typing. It balances fine-grained classification accuracy and system robustness, making it well-suited for real-world information extraction applications.

Zero-Shot Hierarchical Entity Typing Mechanism In this multi-agent framework, the core mechanism for hierarchical entity typing from broad categories like Person to subtypes like "/person/artist" and further to "/person/artist/actor" is realized through the zero-shot reasoning capabilities of the agents. The decision-making process is structured as follows: Each agent is assigned to a specific fine-grained type (typically at the third hierarchical level, such as "/person/artist/actor") and is provided with a detailed description. When encountering unseen entities, agents do not perform a simple binary classification (yes/no). Instead, they engage in a stepwise hierarchical reasoning process. The prompt given to each agent includes an explicit domain definition, for example:

```
"You are a specialist in identifying '/
    person/artist/actor' entities (
    actors in film, television, theater,
     or other media)."
```

This prompt design implicitly encodes the hierarchical dependency. In order to determine whether an entity is an actor, the agent must first verify if it is a person and then if it qualifies as an artist. Leveraging its pre-trained knowledge, the language model understands these inheritance relationships, such as all actors being artists and all artists being persons.

Stepwise Reasoning Process: There are three levels: First-Level: Determine whether the entity is a person, location, organization, or other. Second-Level: If classified as a person, further assess whether it belongs to a subtype such as an artist or athlete. Third-Level: If an artist is classified as such, determine whether it specifically refers to an actor, author, etc. When the agent determines that the entity does not belong to its specialized type, it provides alternative type suggestions, reflecting the hierarchical reasoning process. For example, an actor specialist might respond:

```
"This is not an actor, but it may be a
    '/person/artist/director'."
or
"This is not an actor, and may not even
    be an artist, but it could be a '/
    person/athlete'."
```

C.2 DocRed RE

We conduct experiments on the test set of DocRed (Yao et al., 2019), introducing a novel application of multi-agent collaboration and debate mechanisms for document-level relation extraction. Specifically, we create a dedicated agent for each relation type (e.g., P17 "country", P19 "place of birth"), where each agent focuses solely on identifying its assigned relation, thereby improving relation-specific prediction accuracy. During the multi-round debate process, all agents first independently analyze entity pairs and make their initial predictions. The agents then share their observations and adjust their decisions based on feedback from other agents. Through iterative interactions, the agents gradually reach more stable judgments. In the final consensus stage, we apply a weighted voting mechanism that aggregates agent decisions based on their confidence scores and the number of supporting votes, leading to more reliable relation predictions.

Table 10 shows results on the test set of DocRED under the zero-shot setting using GPT-3.5 and GPT-4. MAF-IE consistently outperforms all baselines on GPT-3.5, achieving a 4.49% improvement over multiagent baseline and surpassing Semi-automatic data enhancement (Li et al., 2023b) methods by 19.17%, 12.95%, and 13.04%, respectively.

C.3 Clinical MACCROBAT-EE

We conduct experiments with GPT-3.5 on the test set of clinical MACCROBAT-EE (Ma et al., 2023), following the same settings used for ACE05, including Event Detection (ED), Event Argument Extraction (EAE), and Event Extraction (EE). As shown in Table 9, prompting GPT-3.5 performs poorly on the clinical MACCROBAT-EE dataset, with near-zero F1 scores on ED and EE and only moderate results on EAE. While the existing multi-agent framework (Subramaniam et al., 2025) improves ED, it underperforms on EAE and EE. In contrast, MAF-IE achieves the best performance across all tasks, with F1 scores of 25.95% (ED), 32.18% (EAE), and 24.45% (EE), demonstrating superior generalizability and robustness in zero-shot event extraction.

C.4 Results for BC5CDR and NYT

We conduct experiments with GPT-3.5 on the NYT (Zeng et al., 2018) test set for the RE task and the BC5CDR (Li et al., 2016) test set for the NER task,

Metric	F1 (%)
Improved F1	4.91
Entity Type	**Improved / Total (%)**
PER	4 / 102 (3.92)
LOC	4 / 102 (3.92)
ORG	7 / 102 (6.86)

Table 6: Improvement statistics on CONLL04 NER, with GPT-3.5

Debate Rounds	Number of Improvements
1 Round	3
2 Rounds	4
4 Rounds	1

Table 7: Incremental improvements across debate rounds on CoNLL04 NER with GPT-3.5.

following the same experimental settings as used on CONLL04 for each corresponding task in zero-shot setting with GPT-3.5. Table 14 presents F1 scores on the NYT RE task under the zero-shot setting. The One-step method of achieves an F1 of 10.5%, showing limited relation extraction capability. Their G&O strategy improves the score to 16.0% by incorporating a generation and refinement process. In contrast, our proposed MAF-IE achieves the best F1 of 19.0%, demonstrating the effectiveness of our multi-agent collaboration framework in enhancing relation extraction across domains. Table 15 presents the F1 scores on the BC5CDR dataset for the NER task in the zero-shot setting. The All-Entity-in-One and One-step achieve F1 scores of 50.58% and 60.41%, respectively. Their G&O strategy further improves the performance to 61.86%. In comparison, MAF-IE achieves the highest F1 score of 64.23%, demonstrating superior effectiveness in zero-shot biomedical NER.

D Prompt Details

D.1 Detail prompts for NER

Listing-1:Type agent w/o debate

```
<Human>Given the following text, extract
     all named entities of the following
     types: Person, Organization,
    Location.
For each extracted entity, provide:
- The entity type (Person, Organization,
     or Location)
Text: {text}

<bot> Response:
```

Contrastive Models	Precision	Recall	F1
Model 1	53.15	75.46	62.37
Model 2	53.26	76.83	62.11
Model 3	53.01	74.77	62.04
Majority Voting	52.58	77.06	**62.51**

Table 8: Majority voting inference with contrastive models on CoNLL04 NER, GPT-3.5.

Method	ED	EAE	EE
GPT-3.5			
E&IO	0	0	0
E&IO	0	29.5†	0
DICE (Ma et al., 2023)			
- E&IO	0	0	0
- Task Inst.	8.37	-	-
CODE4STRUCT (Wang et al., 2023b)	-	11.89	-
Multi-agent framework			
MAFT (Subramaniam et al., 2025)	22.64†	20.33	7.23
MAF-IE			
- All Type-agent	22.28	24.14	15.72†
- Multi-agent Collaboration	**25.95**	**32.18**	**24.45**

Table 9: F1 scores (%) on Clinical MACCROBAT for ED, EAE and EE tasks under different baselines and collaboration frameworks in zero-shot setting. Bold indicates the best performance. † marks the second-best.

In the prompts, entity types are rephrased to enhance model comprehension. For example, “PER” is rewritten as “person”, and “ORG” as “organization”, improving clarity while ensuring consistency across models. Each type’s ontology definition is a key distinguishing feature of its dedicated Type Agent.

Listing-2: MAF-IE

```
Extract all person (PER), location (LOC)
    , and organization (ORG) entities
    from the following text.
As a {self.agent_type} entity
    recognition expert, you should be
    particularly focused on correctly
    identifying all {self.agent_type}
    entities.
Please provide your answer in the
    following format:
PER: ###[list of person entities]###
LOC: ###[list of location entities]###
ORG: ###[list of organization entities
    ]###

If a category has no entities, use 'NULL
    ' inside the ### markers.
Make sure each entity is clearly
    separated by commas within the ###
    markers.

CONFIDENCE: [1-10] - Please provide an
```

Metrics (→) **Baselines (↓)**	**Paradigm**	**F1** (%)
Single model		
GPT-3.5 (OpenAI, 2023a)		
Semi-automatic Data Enhancement for Doc-level (Li et al., 2023b)		
+ GPT-3.5 only	zero-shot	5.6
+ GPT-3.5 only+NLI (w/o. rel des)	zero-shot	11.82
+ GPT-3.5 only+NLI (w. rel des)	zero-shot	11.73
LMRC (Li et al., 2024a)		
+ GPT-3.5 only	3-shot	6.97
+ LMRC	3-shot	10.71
+ Renerta	3-shot	10.71
Multi-agent framework		
+ MAFT (Subramaniam et al., 2025)	zero-shot	<u>20.28</u>†
+ **MAF-IE** (Type-agent w/o debate)	zero-shot	5.98
+ **MAF-IE** (Multi-agent Collaboration)	zero-shot	**24.77**
GPT-4 (OpenAI, 2024b)		
+ Multi-dimensional Prompting (Zhu et al., 2024)	zero-shot	15.58
+ MDP (Zhu et al., 2024)	zero-shot	15.58
+ LMRC (Li et al., 2024a)	3-shot	36.20

Table 10: Main results on DocRed for long-document RE task in zero-shot setting.Bold indicates the best performance and † marks the second-best in zero-shot setting.

```
    overall confidence score for your
    entity identifications

After providing the entities in the
    format above, you can explain your
    reasoning.

Text: {text}
```

D.2 Detail prompts for RE

Listing-3:Type agent w/o debate

```
<Human>Given the following text, extract
     all entites of the following
    relaiton types: Organization based
    in, Located-in, Live-in, Work-for
    and Kill.
Return the extracted relations in the
    following format:
head entity, relation type, tail entity.
Text: {text}
<bot> Response:
```

Listing-4: MAF-IE

```
You are a specialized agent that only
    extracts '{relation_type}'
    relationships from text.

{relation2prompt[relation_type]}

In '{relation_type}' relationships:

Head entity is a {head_type} type

Tail entity is a {tail_type} type

The relationship means the head {
    relation_verb} the tail

IMPORTANT FORMAT: Use exactly this
    format for each relation you find:
Relation: {relation_type}, Head: ###[{
    head_type}]###, Tail: @@@[{tail_type
    }]@@@

For example:
Relation: {relation_type}, Head: ###John
     Smith###, Tail: @@@New York City@@@

If no '{relation_type}' relationships
    are found in the text, explicitly
    state 'No {relation_type}
    relationships found.'

Text: {context}
```

E Additional ablation studies

Performance of small Language Models on different IE We explore the direct prompting performance of Qwen2.5(3B) and Phi-4-mini(4B) on test set of ACE05 (ED,EAE) and CONLL04 (NER,RE) in zero-shot setting.

Table 12 presents the zero-shot performance of small language models on the EAE and ED tasks from the ACE05 test set, as well as the NER and RE tasks from the CONLL04 test set, using direct prompting. For the EAE task, Qwen2.5(3B) significantly outperforms Phi-4-mini(3B), achieving an F1 score of 20.42%, nearly twice that of Phi-4-mini. This improvement is largely attributed to Qwen2.5's higher recall, suggesting its stronger ability to identify event arguments in a zero-shot setting. Nevertheless, both models exhibit low precision, underscoring the inherent challenge of zero-shot EAE for small models. In the ED task, Qwen2.5 (3B) again surpasses Phi-4-mini (3B) with a notable margin (39.96% vs. 21.83% F1), demonstrating more accurate detection of event triggers without prior supervision.

For NER, Qwen2.5 achieves an F1 score of 17.75%, outperforming Phi-4-mini's 12.15%. However, both models fall short compared to larger LLMs, highlighting the challenge of zero-shot NER for small language models with limited capacity.

Interestingly, in the RE task, Phi-4-mini (3B) slightly outperforms Qwen2.5 (3B), achieving 10.76% F1 compared to Qwen2.5's 4.13%. This suggests that while Qwen2.5 excels in argument extraction and event detection, its relational reasoning capabilities under zero-shot prompting may be less robust than Phi-4-mini for this task.

We further analyze the performance of the two small language models on the NER task across

Few-shot Method	Precision	Recall	F1
5-shots	56.54	81.05	66.61
10-shots	57.73	82.06	67.78
15-shots	57.24	82.06	67.44

Table 11: The results of few-shot learning on CoNLL04 NER task with GPT-3.5.

Model	EAE	ED	NER	RE
Qwen2.5(3B)	20.42	39.96	17.75	4.13
Phi-4-mini(3B)	11.02	21.83	12.15	10.76

Table 12: F1 score (%) on EAE, ED, NER, and RE tasks using different small language models.

different entity types. As shown in Table 13, Qwen2.5 (3B) consistently outperforms Phi-4-mini (3B), achieving a higher overall F1 score of 17.75% compared to 12.16%. This improvement is primarily attributed to Qwen2.5's superior recall on PER and LOC entities, where it demonstrates stronger capability in identifying PER and LOC names in a zero-shot setting. However, both models exhibit weak performance on ORG entities, with F1 scores of only 8.27% (Qwen2.5) and 8.95% (Phi-4-mini). This suggests that small language models struggle to recognize ORG names without task-specific adaptation. One possible reason is that ORG entities tend to be more ambiguous and diverse, often containing abbreviations, generic terms, or domain-specific expressions, which are harder to identify without prior fine-tuning.

Overall, these results highlight the limitations of small language models in zero-shot NER, especially for more complex types.

Final answer generation from multiple fine-tuned models via majority voting Table 8 shows that majority voting not only improves the overall F1 score to 62.51%, outperforming all individual models, but also enhances recall while maintaining comparable precision, demonstrating its effectiveness in boosting the collective performance beyond that of each finetuned model.

F Few-shot learning

Table 11 shows the results of few-shot learning on the CONLL04 NER task using GPT-3.5. We observe that increasing the number of provided examples from 5 to 15 does not lead to consistent improvements in F1 score. While there is a slight gain from 5-shot to 10-shot, the performance plateaus or even slightly drops afterward. This suggests that simply adding more ground-truth examples in the prompt reaches a saturation point, beyond which the model struggles to further benefit from additional examples. One possible reason is the model's limited capacity to generalize from few-shot prompts, as it tends to memorize surface patterns without fully understanding the underlying task structure. This observation highlights the limitations of prompt-based few-shot learning with large language models for structured prediction tasks like NER.

Models	Precision	Recall	F1
Qwen2.5(3B)			
- PER	18.99	41.29	26.02
- LOC	15.93	29.39	20.67
- ORG	4.86	27.66	8.27
Overall	12.10	33.33	17.75
Phi-4-mini(3B)			
- PER	7.53	11.34	9.05
- LOC	15.05	21.96	17.86
- ORG	5.62	21.99	8.95
Overall	9.15	18.13	12.16

Table 13: Precision, Recall, and F1 (%) on NER task of different small language models.

Method	F1
GPT-3.5	
G&O (Li et al., 2024b)	
- One-step	10.5
- G&O	<u>16.0</u>†
MAF-IE	**19.0**

Table 14: F1 scores (%) of GPT-3.5 on NYT for RE task under different baselines in zero-shot setting.

G Mixture of agents

We notice that recent work **Mixture-of-Agents** (Wang et al., 2024b) combines multiple powerful LLMs (e.g., Qwen2.5-70B-Instruct (Simonycl, 2024), Llama3.1-70B-chat (AI, 2024), Qwen1.5-110B-chat (Team, 2024a) and WizardLM-8x22B (Alpindale, 2024)) as heterogeneous agents to leverage their complementary strengths for collaborative task solving. However, this approach is fundamentally different from ours: rather than relying on cross-model complementarity, MAF-IE focus is on improving a base model through multi-agent fine-tuning, enabling a more scalable and lightweight training paradigm.

Method	F1
GPT-3.5	
G&O (Li et al., 2024b)	
- All-Entity-in-One	50.58
- One-step	60.41
- G&O	61.86†
MAF-IE	**64.23**

Table 15: F1 scores (%) of GPT-3.5 on BC5CDR for NER task under different baselines in zero-shot setting.

Metric	ACE05	ERE	MACCROBAT-EE
Unique event types	33	38	13
Unique argument roles	22	21	22
Unique arg. roles per event type	4.73	2.87	10
Documents #	599	459	200
Sentences #	20,862	17,114	4,539
Entities #	54,820	46,185	23,898
Trigger mentions #	5,348	7,287	13,128
Argument mentions #	8,102	10,479	8,599
Avg entities # per sentence	3.18	3.20	5.43
Avg events # per sentence	1.34	1.47	3.21
Avg args # per sentence	2.39	2.24	2.67
Avg args per event #	1.48	1.42	0.81
Avg entity word count	1.12	1.10	1.89
Avg trigger word count	1.05	1.06	1.61
Avg argument word count	1.14	1.14	1.72

Table 16: Statistics of ACE05, ERE, and MACCROBAT-EE datasets.

H Time and cost efficiency

Tables 23 and 24 analyze the trade-offs between performance, inference time, and cost across different strategies on CONLL04 NER and RE tasks with GPT-3.5.

H.1 Time Efficiency

As shown in Table 23, single-agent inference, whether using the base GPT-3.5 model or its fine-tuned variant, achieves the fastest inference time of 12.5 seconds per sample, leveraging the absence of multi-agent interactions. In contrast, multi-agent debate introduces significant latency overhead. Specifically, 3-agent debate on NER takes 21.5 seconds per instance (72% increase), while 5-agent debate on RE takes 25.0 seconds per instance, reflecting the increasing latency with larger agent groups and deeper interactions. Notably, multi-agent parallel inference after fine-tuning brings the latency back to 12.5 seconds, matching single-agent inference. This is achieved by parallel execution of multiple fine-tuned agents without iterative debating, making it significantly more time-efficient compared to multi-agent debate.

Description	Dev	Test
Candidate Space	395,572	392,158
# NA Entity Pairs	384,949	-
# Relation Entity Pairs	10,623	-
# Annotated Triples	12,275	-

Table 17: Statistics of DocRED.

Description	Train	Dev	Test
# Sentences	910	243	288
avg. l-text	-	-	159
n-ner-type	-	-	3
n-relation-type	-	-	5
n-ary-relations	-	-	2
n-relation-mention	-	-	422

Table 18: Statistics of CoNLL04. "n-ary-relations" indicates the number of entities in a relation tuple (group).

H.2 Cost Efficiency

As shown in Table 24, single-agent inference also achieves the lowest cost of \$0.000336 per instance, leading to the highest Efficiency Score on both NER (191,101) and RE (103,333). While multi-agent debate improves F1 (e.g., from 64.21% to 66.83% on NER), it increases the cost to \$0.000841 (NER, 3 agents) and \$0.001682 (RE, 5 agents), substantially lowering the Efficiency Score (NER: 79,465; RE: 21,686). In comparison, fine-tuned multi-agent parallel inference maintains strong F1 (NER: 63.65%, RE: 33.47%) while reducing cost by 40%-60% compared to multi-agent debate (NER cost: \$0.001008 vs. \$0.000841, RE cost: \$0.001680 vs. \$0.001682), resulting in better cost-effectiveness than debate (NER: 63,179; RE: 19,924).

Summary These findings demonstrate that fine-tuned multi-agent parallel inference offers a superior balance of performance, time, and cost. It retains much of the accuracy gain from multi-agent collaboration while eliminating the time and cost overhead associated with multi-round debates. This makes it a more practical and scalable choice for real-world deployment.

Efficiency Score metric We were inspired by prior work on computational efficiency in NLP models (Strubell et al., 2019; Kaplan et al., 2020) and calculate the efficiency score as follows:

$$\text{Efficiency Score} = \frac{\text{F1-score}}{\text{Cost Per Doc\_ID}}.$$

Dataset	OntoNotes
# of Types	89
# of Documents	300k
# of Entity Mentions	242k
# of Train Mentions	223k
# of Test Mentions	8963

Table 19: Statistics of OntoNotes.

Dataset	Domains	Docs	Ent	Rel	Trig	Arg
ACE05-E	News	599	7	-	33	22

Table 20: Statistics of ACE05-E. Ent: Number of entity categories. Rel: Number of relation categories. Trig: Number of event trigger categories. Arg: Number of event argument categories.

Dataset	BC5CDR
n-instance	1,000
avg. l-text	148
n-entity-type	2
n-entity-mention	2,074

Table 21: Statistics of BC5CDR. "avg. l-text" denotes the average number of characters in each text instance.

Dataset	NYT
n-instance	369
avg. l-text	199
n-relation-type	7
n-ary-relations	2
n-relation-mention	265

Table 22: Statistics of NYT. "n-ary-relations" indicates the number of entities in a relation tuple (group).

I Additional Case study

Comparative error analysis against the baseline Our error analysis in Table 6 shows that our type-specialized multi-agent debate and finetuning framework achieves consistent improvements across all entity types on the CONLL04 NER task with GPT-3.5 compared with prior work MAFT (Subramaniam et al., 2025), yielding an overall F1 increase of 4.91%. Specifically, we observe the largest gain on ORG entities (+6.86%), followed by PER and LOC (+3.92% each).

We attribute these improvements to the unique strengths of our multi-agent framework. First, the type-specialized agents promote targeted extraction by focusing on entity-specific decision boundaries. This is particularly beneficial for complex types like ORG that often suffer from boundary ambiguity and semantic overlap with other types in single-model settings. By contrast, single-model baselines tend to produce generalized predictions without type-specific refinement, limiting their ability to distinguish challenging cases.

Second, our cross-agent verification and debate mechanism encourages agents to reflect on their initial outputs, enabling error correction through collaborative reasoning. This is especially effective for resolving missed or misclassified entities, as agents are required to justify and revise their predictions based on structured prompts and peer feedback. The observed improvements for PER and LOC suggest that this iterative refinement process helps recover subtle mentions easily overlooked in single-pass predictions.

Finally, adopting lightweight majority voting during inference mitigates the risk of overfitting or output homogenization introduced by excessive multi-round debate. By aggregating independent predictions from specialized agents, our framework balances diversity and consistency, leading to more robust extractions with minimal computational overhead.

These findings highlight the effectiveness of integrating type specialization, collaborative reasoning, and lightweight voting in improving overall and type-specific extraction performance. They also suggest future opportunities to further enhance our framework by incorporating task-adaptive debate strategies or confidence calibration techniques to handle entity types with high contextual variability better.

Representative case studies Additionally, as shown in Table 25, we present five representative cases where our proposed framework achieves notable F1 improvements compared to prior work MAFT (Subramaniam et al., 2025). These examples provide qualitative analysis demonstrating how our type-agent collaborative framework effectively corrects entity recognition errors made by the baseline model. For example, in Document 44, our model successfully identifies "president-elect bush" as a PER, which was previously missed by the baseline. Similar improvements are observed for location entities such as "bosnia" and "german," as well as person entities like "bruce" and "president reagan." These results indicate that our multi-agent system is better at capturing entity boundaries and resolving semantic ambiguities, further validating the effectiveness of our collaborative interaction

Task (↓)	Inference Mode	# Agents	Avg. Latency (s)
CoNLL04 NER			
	Single-Agent Inference	1	12.5
	3-Agent Debate	3	21.5
	Single Finetuned Agent Inference	1	12.5
	3-Agent Parallel Inference (Finetuned)	3	12.5
CoNLL04 RE			
	Single-Agent Inference	1	12.5
	5-Agent Debate	5	25.0
	Single Finetuned Agent Inference	1	12.5
	3-Agent Parallel Inference (Finetuned)	5	12.5

Table 23: Comparison of time efficiency on CoNLL04 NER and RE tasks (average seconds per test sample). Parallel inference achieves the same latency as single-agent inference, while debate significantly increases latency as the number of agents grows.

Task (↓)	Inference Mode	# Agents	F1-score (%)	Cost per Doc_ID (USD)	Efficiency Score
CoNLL04 NER					
	Single-Agent Inference (GPT-3.5)	1	58.15	$0.000336	173,660
	Single-Agent Inference (Finetuned)	1	64.21	$0.000336	191,101
	3-Agent 2-Round Debate	3	66.83	$0.002016	33,166
	3-Agent Parallel Inference (Finetuned)	3	63.65	$0.001008	63,179
CoNLL04 RE					
	Single-Agent Inference (GPT-3.5)	1	34.72	$0.000336	103,333
	Single-Agent Inference (Finetuned)	1	28.63	$0.000336	85,208
	5-Agent 2-Round Debate	5	36.47	$0.003360	10,850
	5-Agent Parallel Inference (Finetuned)	5	33.47	$0.001680	19,924

Table 24: Cost efficiency comparison on CoNLL04 NER and RE tasks. Efficiency Score is calculated as F1-score divided by Cost per Doc_ID (USD). We set the debate rounds to two.

design for specific IE tasks.

Stepwise impact of debate Furthermore, Table 7 analyzes how the number of debate rounds affects performance improvements. Our results show that most gains are achieved within the first one or two rounds, while the benefits of additional rounds gradually diminish. Notably, only one improvement is observed after four rounds, suggesting that increasing the number of debate rounds may lead to diminishing returns. This finding indicates that early-stage agent collaboration is generally sufficient to resolve most disagreements and correct recognition errors, whereas excessive rounds may introduce noise or redundant reasoning.

Case 1: PER Entity (Doc ID: 44, F1 Gain: +0.3333)

Misclassified Entity: president-elect bush
Context: "These are the tactics of a marginalized force driven to extremes by desperation, said Abram..."

Case 2: LOC Entity (Doc ID: 146, F1 Gain: +0.3333)

Misclassified Entity: bosnia
Context: "BSP, SDS Support Noninvolvement in Bosnia AU1502173794 Sofia BTA in English 1646 GMT 15 Feb 94..."

Case 3: LOC Entity (Doc ID: 162, F1 Gain: +0.3333)

Misclassified Entity: german
Context: "Esprit Project to Develop Chip to Receive, Transmit Nerve Impulses..."

Case 4: PER Entity (Doc ID: 264, F1 Gain: +0.3333)

Misclassified Entity: bruce
Context: "Springsteen, a New Jersey native, was clearly the favorite..."

Case 5: PER Entity (Doc ID: 36, F1 Gain: +0.2000)

Misclassified Entity: president reagan
Context: "Also under consideration are two conservative federal appellate judges appointed by President Reagan..."

Table 25: Top improvements with example cases.

Algorithm 1: MAF-IE NER using type-specialized agents

Input: Docs D, Agent model A, Max rounds M, Consensus threshold θ (default: $2/3$)
Output: Consensus entities for each document

Init experts $A_{PER}, A_{LOC}, A_{ORG} \leftarrow A$;
for *each doc* $d \in D$ **do**
    Round 0: each expert A_t outputs initial entities $e_{t,0}$;
    Store $E_0 \leftarrow \{e_{PER,0}, e_{LOC,0}, e_{ORG,0}\}$;
    for $m = 1$ **to** M **do**
        for *each type* $t \in \{PER, LOC, ORG\}$ **do**
            $others \leftarrow$ results from other experts in round $m - 1$;
            $e_{t,m} \leftarrow A_t(d, others, e_{t,m-1})$;
        end
        $E_m \leftarrow \{e_{PER,m}, e_{LOC,m}, e_{ORG,m}\}$;
    end
    for *each type* $t \in \{PER, LOC, ORG\}$ **do**
        $final_t \leftarrow$ [$e_{t,M}$ from each expert];
        $conf_t \leftarrow$ [each expert's confidence];
        $weights_t \leftarrow$ [2.0 if expert specializes in t, else 1.0];
        $votes_t \leftarrow$ calculate the weighted vote sum for each entity;
        $cons_t \leftarrow$ [entity | votes(entity) $\geq$ total_experts $\times\theta$];
    end
    $consensus \leftarrow \{cons_{PER}, cons_{LOC}, cons_{ORG}\}$;
    if *ground truth* g_d *available* **then**
        $metrics_d \leftarrow evaluate(consensus, g_d)$;
    end
    return $consensus, metrics_d$;
end

Algorithm 2: MAF-IE RE using type-specialized agents

Input: Docs D, Relations $\mathcal{R}$, Agent model A, Agents/relation k, Max rounds M, Threshold θ
Output: Relations for each document

```
Init experts $\{A_r^1, ..., A_r^k\}$ for each $r \in \mathcal{R}$;
for each doc $d \in D$ do
    for each $r \in \mathcal{R}$ do
        Round 0: Each expert $A_r^i$ extracts $e_r^{i,0}$;
        Store $E_r^0 \leftarrow \{e_r^{1,0}, ..., e_r^{k,0}\}$;
    end
    for $m = 1$ to $M$ do
        for each $r \in \mathcal{R}$ do
            for $i = 1$ to $k$ do
                $input \leftarrow$ results from $\{A_r^j\}_{j \neq i}$ in round $m - 1$;
                $e_r^{i,m} \leftarrow A_r^i(d, input, e_r^{i,m-1})$;
            end
            $E_r^m \leftarrow \{e_r^{1,m}, ..., e_r^{k,m}\}$;
        end
    end
    $results \leftarrow \{\}$;
    for each $r \in \mathcal{R}$ do
        $votes \leftarrow$ count for each extracted relation;
        $cons_r \leftarrow$ [rel | votes(rel) $\geq k \times \theta$];
        $results \leftarrow results \cup cons_r$;
    end
    if ground truth $g_d$ available then
        $metrics_d \leftarrow evaluate(results, g_d)$;
    end
    return $results, metrics_d$;
end
Compute P/R/F1 over all documents;
Compute metrics for each relation type;
```

Algorithm 3: CONTRASTIVE DATA PREPARATION for Multi-Agent NER

Input: Consensus dir D_c, Initial preds dir D_i, Types $\mathcal{T}$
Output: Training data for critic fine-tuning

$c\_ex \leftarrow$ [{} for each type]; $ic\_ex \leftarrow$ [{} for each type];
$c\_cnt \leftarrow$ [0,0,0]; $ic\_cnt \leftarrow$ [0,0,0];
for *each file* $f \in D_c$ **do**
    $id \leftarrow$ extract doc ID from f;
    $ctx, c\_ent \leftarrow$ load from file f;
    for *each type* $t \in \mathcal{T}$ **do**
        $idx \leftarrow$ get index for type t;
        $i\_file \leftarrow D_i/doc\_\{id\}\_\{t\}\_initial.json$;
        if i_file *exists* **then**
            $i\_ent, m\_resp \leftarrow$ load from i_file;
            $prompt \leftarrow$ construct with ctx and m_resp;
            $correct \leftarrow$ compare i_ent with c_ent;
            if $correct$ **then**
                $resp \leftarrow$ construct positive feedback;
                $c\_ex[idx][c\_cnt[idx]] \leftarrow [prompt, resp]$;
                $c\_cnt[idx] \leftarrow c\_cnt[idx] + 1$;
            else
                $resp \leftarrow$ construct criticism;
                $ic\_ex[idx][ic\_cnt[idx]] \leftarrow [prompt, resp]$;
                $ic\_cnt[idx] \leftarrow ic\_cnt[idx] + 1$;
            end
        end
    end
end
for $i = 0$ **to** 2 **do**
    $c\_data \leftarrow$ list items from $c\_ex[i]$;
    $ic\_data \leftarrow$ list items from $ic\_ex[i]$;
    Shuffle both datasets;
    $train \leftarrow$ merge ic_data with balanced c_data;
    Save $train$ to file (JSON format);
end
return training datasets;

Algorithm 4: INFERENCE with Majority Voting

Input: Docs D, Models $M = \{m_1, m_2, ..., m_k\}$, Voting threshold θ

Output: Entity predictions and performance metrics

$results \leftarrow \{\}$;
$metrics \leftarrow$ initialize metrics counters;
for *each doc* $d \in D$ **do**
    $context \leftarrow$ text content of d;
    $gt \leftarrow$ ground truth entities of d;
    $model\_ents \leftarrow []$;
    $model\_metrics \leftarrow []$;
    for *each model* $m \in M$ **do**
        $prompt \leftarrow$ create NER prompt with $context$;
        $response \leftarrow$ generate using model m with $prompt$;
        $entities \leftarrow$ extract PER, LOC, ORG from $response$;
        Add $entities$ to $model\_ents$;
        $metric \leftarrow$ calculate precision, recall, F1 between $entities$ and gt;
        Add $metric$ to $model\_metrics$;
        Update global metrics for model m;
    end
    $votes \leftarrow$ count entity occurrences across all models;
    $consensus \leftarrow \{\}$;
    for *each entity type* $t \in \{PER, LOC, ORG\}$ **do**
        $consensus_t \leftarrow []$;
        for *each entity* e *with type* t **do**
            if *votes(e)* $\geq |M| \times \theta$ **then**
                Add e to $consensus_t$;
            end
        end
        $consensus[t] \leftarrow consensus_t$ or ["NULL"] if empty;
    end
    $mv\_metrics \leftarrow$ calculate metrics between $consensus$ and gt;
    Update global majority vote metrics;
    Store document results in $results$;
end
Calculate final precision, recall, F1 for each model;
Calculate final precision, recall, F1 for majority vote;
Create comparative performance tables;
return $results$, performance metrics;