CrossAgentIE: Cross-Type and Cross-Task Multi-Agent LLM Collaboration for Zero-Shot Information Extraction

Meng Lu[♠], Yuzhang Xie[♡], Zhenyu Bi[♠], Shuxiang Cao[◇], Xuan Wang[♠*]

[♠]Department of Computer Science, Virginia Tech, Blacksburg, VA, USA
[♡]Department of Computer Science, Emory University, GA, USA
[◇]Department of Physics, University of Oxford, Oxford, UK

(mengl,zhenyub,xuanw)@vt.edu; yuzhang.xie@emory.edu; shuxiang.cao@physics.ox.ac

Abstract

Large language models (LLMs) excel in generating unstructured text. However, they struggle with producing structured output while maintaining accuracy in zero-shot information extraction (IE), such as named entity recognition (NER) and relation extraction (RE). To address these challenges, we propose CrossAgentIE, a multi-agent framework that enhances zero-shot IE through multi-agent LLM collaboration. CrossAgentIE refines LLM predictions iteratively through two mechanisms: intra-group cross-type debate, which resolves entity-label conflicts through context-based evidence and confidence aggregation, and inter-group cross-task debate, where NER and RE mutually refine outputs via bidirectional feedback. Furthermore, we introduce template fine-tuning, distilling high-confidence multi-agent outputs into a single model, significantly reducing inference costs while preserving accuracy. Experiments across five NER and five RE datasets show that CrossAgentIE significantly outperforms state-of-the-art zero-shot baselines by a large margin. CrossAgentIE effectively addresses LLM limitations in structured prediction with an effective and efficient approach for zero-shot information extraction. Our GitHub can be found at https://github.com/Luca/CorssAgentIE.

1 Introduction

Information extraction (IE) is a fundamental task in natural language processing (NLP) that aims to extract structured information from unstructured or semi-structured text (Li et al., 2023; Lu et al., 2022). It includes subtasks such as named entities recognition (NER) and relation extraction (RE). Traditional supervised IE methods typically follow a "pre-training → fine-tuning" paradigm, where a pre-trained language model is adapted to a labeled dataset with extensive supervision signals (Devlin et al., 2019; Raffel et al., 2023; Zhuang et al., 2021). While effective, these methods suffer from high annotation costs and limited generalization, making them impractical for low-resource scenarios and rapidly evolving domains.

Given these limitations, recent research has explored zero-shot IE as a promising direction (Wei et al., 2024). Recent advances in large language models (LLMs) (Lin et al., 2023; OpenAI, 2023b) have enabled more effective zero-shot IE methods to overcome the shortcomings of traditional supervised models. The LLMs' strong language understanding capabilities, gained through extensive pre-training, allow them to perform IE tasks effectively. LLM-based approaches for zero-shot IE include direct prompting (Han et al., 2024; Wang et al., 2023b; Xie et al., 2023a), in-context learning (Brown et al., 2020; Min et al., 2022), synthetic data generation (Heng et al., 2024), and pseudo-labeling for fine-tuning (Gao et al., 2024a; Heng et al., 2024; Sainz et al., 2024; Zaratiana et al., 2024). These methods reduce reliance on annotated data and enhance adaptability, making LLMs a promising solution for zero-shot IE.

Despite advancements, LLMs still encounter critical challenges that limit their performance in zero-shot IE. **First**, LLMs struggle to generate structured outputs that adhere to predefined labeling schemas in IE. Unlike traditional models optimized for structured representations, LLMs predominantly generate free-form text. Although prompting techniques, such as using symbols (Wang et al., 2023b), lists (Zhou et al., 2024), and tables (Jiao et al., 2023), have been explored, inconsistencies persist in structured output generation. **Second**, entity-label conflicts arise when identical entities receive inconsistent categorizations (e.g., "Washington" might be labeled as both *Location* and *Person*). Existing approaches (Li et al., 2024a; Heng et al., 2024) tackle this issue through weak supervision, either by fine-tuning smaller models

[*]Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2025, pages 13953–13977
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

on pseudo-labeled data or by transferring knowledge from limited annotations. However, they rely on external supervision rather than leveraging the intrinsic reasoning embedded in LLMs' representations, which limits the generalization of these methods to broader scenarios. **Third**, LLMs struggle with domain adaptation, failing to internalize domain-specific knowledge despite task instructions. While prompt engineering can create role-specialized agents (Lu et al., 2024; Wang and Huang, 2024), these methods require extensive tuning and lack cross-domain generalization. As a result of the above three challenges, current LLM-based methods struggle with achieving high performance in zero-shot IE (Jiang et al., 2024b; Shen et al., 2023; Wan et al., 2023). For example, direct prompting with GPT-3.5 achieves only 45% F1 on CoNLL03 (Li et al., 2024a) and 34% on OntoNotes4 (Xie et al., 2023a) for NER.

To address the above challenges, we propose CROSSAGENTIE, a multi-agent LLM collaboration framework that enhances zero-shot NER and RE performance through structured debate and bidirectional refinement. First, intra-group cross-type debate resolves entity-label conflicts by verifying classifications (e.g., distinguishing "Washington" as "Location" or "Person") through context-based reasoning. Second, inter-group cross-task debate refines NER and RE predictions by integrating relation-based feedback, enhancing contextual grounding, and entity accuracy through bidirectional knowledge exchange. Third, to enhance domain adaptation, CROSSAGENTIE equips type agents with domain-specific metadata, leveraging entity-type knowledge and ontology constraints for schema-aligned classification. Finally, to improve inference efficiency, CROSSAGENTIE introduces template fine-tuning that distills the multi-agent outputs into a single model. This process reduces computational cost while ensuring cross-domain consistency, greatly enhancing the efficiency of CROSSAGENTIE. Experiments across different datasets show that CROSSAGENTIE significantly outperforms state-of-the-art zero-shot baselines by a large margin.

2 Related Work

LLMs for IE Recent advances in LLM-based IE have shown promise in tasks such as NER and RE. NER identifies and classifies entities in unstructured text into predefined categories (Keraghel et al., 2024), while RE extracts relations between entities from the text (Gao et al., 2024b). ChatIE (Wei et al., 2024) enhances IE through structured dialogue with ChatGPT, enabling iterative refinement. InstructUIE (Wang et al., 2023c) employs multi-task instruction tuning to guide LLMs in NER, RE, and event extraction (EE) using natural language prompts. ULTRA (Zhang et al., 2024a) enhances EE with a hierarchical framework, leveraging open-source LLMs for cost-effective extraction while mitigating positional bias.

LLMs for NER Several approaches enhance NER with LLMs. GPR-NER (Wang et al., 2023b) reformulates NER as text generation with entity markers and self-verification, reducing over-predictions via few-shot and in-context learning. UniversalNER (Zhou et al., 2024) distills ChatGPT-generated data into a smaller LLaMA-based model through instruction tuning. VerifiNER (Kim et al., 2024) integrates LLMs with external knowledge bases for post-hoc verification, refining entity boundaries and types. Decomposed-QA (Xie et al., 2023a) improves NER via task decomposition, syntactic augmentation, and self-consistency voting with ChatGPT. ProGen (Heng et al., 2024) uses step-by-step generation and self-reflection to enhance few-shot NER dataset construction and entity attribute refinement.

LLMs for RE Several methods enhance RE with LLMs. GPR-RE (Wan et al., 2023) optimizes GPT's in-context learning via improved example retrieval and reasoning. URE (Wang et al., 2023a)refines relational embeddings using positive pair augmentation, margin loss, and contrastive learning with BERT (Devlin et al., 2019). QA4RE (Zhang et al., 2023) reformulates RE as a multiple-choice QA task, converting relation templates into instruction-tuned options. G&O (Li et al., 2024a) employs a "generation and organization" pipeline for zero-shot RE.

Multi-Agent LLM for IE The rise of LLM-powered agents such as GPTs (Brown et al., 2020; OpenAI, 2023b,a,c), LLaMAs (Touvron et al., 2023), and PaLM (Anil et al., 2023; Chowdhery et al., 2022) has enabled multi-agent collaboration. These systems follow either cooperative strategies to achieve shared goals (Zhang et al., 2024b; Zhou et al., 2023; Qian et al., 2024; Lu et al., 2024), or adversarial strategies to refine outputs (Aryan, 2024; Estornell and Liu, 2024). DAO (Wang and

Huang, 2024) employs a multi-agent optimization framework to refine LLM outputs for EE, integrating external tools to enhance retrieval quality and prediction reliability. Applying multi-agent debate to IE presents challenges such as real-time coordination, entity conflict resolution (Liu et al., 2024), and effective discussion management (Cho et al., 2024). Addressing these challenges enhances IE accuracy, especially in domain-specific contexts.

3 CrossAgentIE Framework

This section introduces CrossAgentIE, a multi-stage framework for structured information extraction using collaborative agents. We first formalize the problem (Sec. 3.1), followed by type-agent setup (Sec. 3.2), intra-group cross-type discussion (Sec. 3.3), inter-group cross-task discussion (Sec. 3.4), and finally template fine-tuning (Sec. 3.5). Figure 1 illustrates the overall framework, with detailed prompts provided in Appendix D.

3.1 Problem Definition

We formalize Named Entity Recognition (NER) and Relation Extraction (RE) as structured information extraction tasks. Given a sentence $s = \{w_1, \ldots, w_n\}$ consisting of n words, the NER task identifies text spans within s as entity mentions and assigns each mention a label from a predefined ontology (e.g., Location, Person). The extracted entity set is denoted as $E = \{e_1, \ldots, e_k\}$, where k is the number of identified entities. Each entity e_i consists of a text span t_i and an entity label l_i, i.e., $e_i = (t_i, l_i)$. Based on E, the RE task extracts a set of relations $R = \{r_1, \ldots, r_m\}$, where m is the number of extracted relations. Each relation $r_i = (e_p, r_i, e_q)$ represents a directed relation r_i between two entities e_p and e_q within E. Additionally, we define a set of collaborative agents $A = \{A_1, A_2, \ldots, A_M\}$, where M denotes the number of agents, which iteratively refine entity recognition and relation extraction results. The final refined entity and relation sets, denoted as E^* and R^*, are obtained through the iterative refinement process: $E^* = f(E, A)$ and $R^* = g(R, A)$, where f and g are refinement functions modeled as interactions among agents.

3.2 Type Agent Setup

To reduce inter-category confusion and improve classification accuracy, we assign each entity and relation type to a specialized agent. Rather than using a single multi-tasking model that processes multiple entity and relation types within a unified framework, each specialized agent makes task-specific decisions with tailored prompting strategies. For instance, NER agents (e.g., PER, LOC) identify entities such as "Reagan" as PER and "America" as LOC, while RE agents (e.g., Live-in) extract head and tail entities based on representative relationships. More details for type agent prompting are in Appendix D.

3.3 Intra-Group Cross-Type Discussion

After setting up the type agents, we introduce a structured debate mechanism to resolve conflicts when multiple agents assign different labels to the same entity. This mechanism enables conflicting agents to engage in discussions and refine their classifications through ontology constraints and contextual reasoning. This process follows a debate-driven iterative refinement framework, where agents engage in multiple debate rounds to reach a consensus. Each type agent A_i^{Type} generates a set of entities S_i^{Type}, with conflicts occurring when agents assign inconsistent labels to the same entity. The conflict set is defined as $C = \{e_i \mid \exists A_j^{\text{Type}}, A_k^{\text{Type}} \text{ such that } l_j(e_i) \neq l_k(e_i), \forall i \in T\}$. During conflict resolution, the agents A_j^{Type} and A_k^{Type} iteratively refine their classifications for each entity $e_i \in C$ by re-evaluating prior classifications, reassessing the entity's context, and enforcing ontology-driven constraints to ensure consistency. If consensus is reached, the entity is assigned a final type. Otherwise, a separate LLM, the Summarizer, aggregates reasoning paths, confidence scores, and contextual evidence to determine the most probable classification. This hybrid approach ensures robust decision-making by combining structured debate resolution with LLM-based consolidation, improving classification accuracy and consistency across entity types.

3.4 Inter-Group Cross-Type Discussion

After resolving conflicts within a single task through intra-group cross-type discussion, we further refine outputs via inter-group cross-task discussion, where NER and RE agents exchange feedback to enhance coherence. At this stage, NER agents generate a candidate set of extracted entities, guiding RE agents to focus on relevant entity types for relation extraction. For example, in the "Live-in" relation, RE agents identify entity pairs consisting of a "Person" and a "Location" (e.g.,

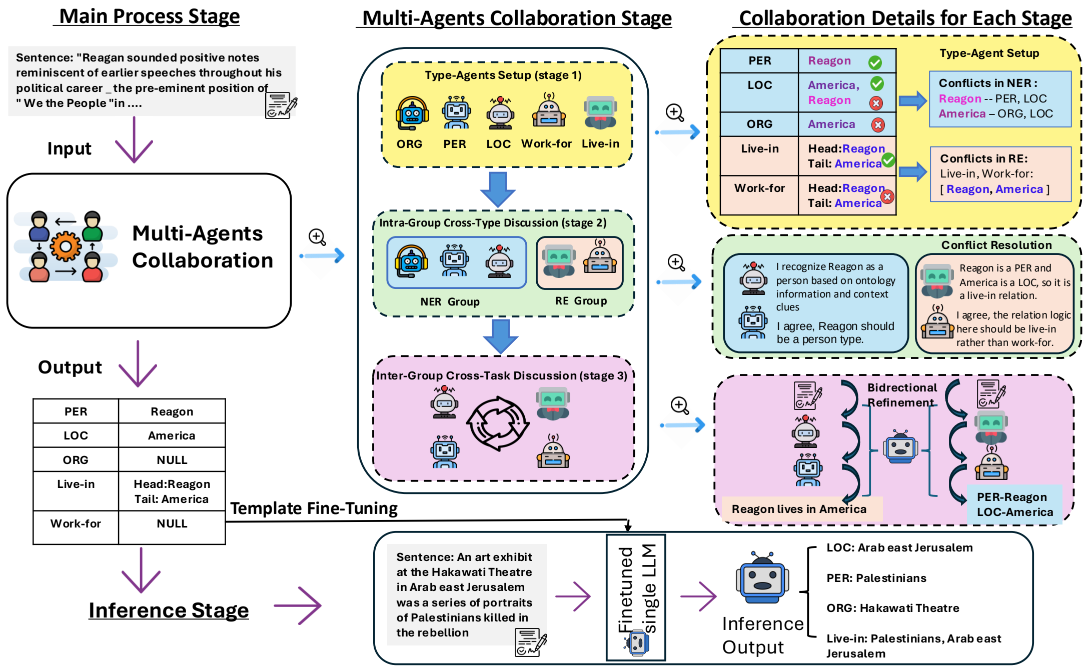
Main Process Stage
Multi-Agents Collaboration Stage
Collaboration Details for Each Stage
Sentence: "Reagan sounded positive notes reminiscent of earlier speeches throughout his political career _the pre-eminent position of " We the People "in ....
Input
Multi-Agents Collaboration
Output
PER | Reagon
LOC | America
ORG | NULL
Live-in | Head:Reagon Tail: America
Work-for | NULL
Template Fine-Tuning
Inference Stage
Type-Agents Setup (stage 1)
ORG PER LOC Work-for Live-in
Intra-Group Cross-Type Discussion (stage 2)
NER Group
RE Group
Inter-Group Cross-Task Discussion (stage 3)
PER Reagon
LOC America, Reagon
ORG America
Live-in Head:Reagon Tail: America
Work-for Head:Reagon Tail: America
Type-Agent Setup
Conflicts in NER : Reagon -- PER, LOC America – ORG, LOC
Conflicts in RE: Live-in, Work-for: [Reagon, America]
Conflict Resolution
I recognize Reagon as a person based on ontology information and context clues
I agree, Reagon should be a person type.
Reagon is a PER and America is a LOC, so it is a live-in relation.
I agree, the relation logic here should be live-in rather than work-for.
Bidirectional Refinement
Reagon lives in America
PER-Reagon LOC-America
Sentence: An art exhibit at the Hakawati Theatre in Arab east Jerusalem was a series of portraits of Palestinians killed in the rebellion
Finetuned single LLM
Inference Output
LOC: Arab east Jerusalem
PER: Palestinians
ORG: Hakawati Theatre
Live-in: Palestinians, Arab east Jerusalem

Figure 1: The overview of CROSSAGENTIE illustrates the multi-agent collaboration process through four stages, converting an input document into structured outputs. The four stages include: 1) Type-agents setup, 2) Intra-group cross-type discussion, 3) Inter-group cross-task discussion, and 4) Template fine-tuning on a single LLM.

PER: Reagan; LOC: America). Afterwards, based on the extracted entities, RE agents generate relation statements (e.g., “Reagan lives in America”) and integrate them into the NER input as contextual knowledge, refining entity classification.

This iterative exchange helps resolve classification ambiguities. In stage 3 of Figure 1, NER agents initially misclassify “America” as both ORG and LOC. However, the “Live-in” relation (i.e., a person must live in a location rather than an organization) enables RE agents to confirm “America” as LOC and provide feedback to NER, prompting the removal of the incorrect ORG label. Similarly, RE agents may initially misclassify “Reagan-America” as both “Live-in” and “Work-for”. Here, NER agents reinforce entity consistency by verifying that “America” is LOC, enabling RE to refine its relation classification.

While this iterative refinement process corrects specific classification errors, a broader challenge remains: how to ensure that NER and RE consistently converge toward a unified entity-relation structure. Since NER and RE operate independently in zero-shot settings, discrepancies naturally arise— NER may extract entities that are irrelevant to RE, while essential entities for RE may be absent from the NER output. To address these inconsistencies, we introduce a mathematical formulation that explicitly quantifies the *symmetric difference* between the entities extracted and required by NER and RE, which is defined as $\Delta(A, B) = (A \setminus B) \cup (B \setminus A)$, where $A = \{NER_{\text{ext}}, RE_{\text{ext}}\}$ represents the entities extracted by NER and RE, and $B = \{NER_{\text{req}}, RE_{\text{req}}\}$ represents the entities required by NER and RE. By minimizing $\Delta(A, B)$, we ensure better alignment between entity boundaries and relation predictions, reducing both spurious and missing entities. The complete mathematical details, including the definition of entity discrepancies, the role of logical constraints, and the minimization of prediction inconsistencies, are provided in Appendix F.

3.5 Template Fine-tuning

After resolving conflicts through intra-group discussion and refining predictions via inter-group interactions between NER and RE, we further optimize inference efficiency. While structured collaboration enhances classification accuracy, its iterative nature incurs substantial computational costs, particularly for multi-label datasets. To mitigate this, we propose template fine-tuning, which distills high-confidence outputs into a single model. By integrating refined results from multiple agents, this approach enhances zero-shot performance on multi-label datasets while preserving accuracy and significantly reducing computational overhead. Please see Appendix J for details.

4 Experiments

We evaluate CrossAgentIE on NER and RE benchmarks using strict full-matching criteria, comparing them with state-of-the-art baselines. As prior work (Section 2) has applied LLMs in various settings, we select the most relevant state-of-the-art (SOTA) zero-shot approaches. See Appendix B for methods comparison. For fair comparison, we use GPT-3.5 as the backbone, aligning with existing baselines, and additionally test our approach on GPT-4o for evaluation on a more advanced LLM. Please see Section 4.3 for details.

4.1 Experimental Setup

NER Datasets We evaluate NER performance on CONLL03 (Tjong Kim Sang and De Meulder, 2003), CONLL04 (Carreras and Màrquez, 2004), OntoNotes4 (Pradhan et al., 2013), Semeval (Hendrickx et al., 2010) and TACRED (Zhang et al., 2017). Please refer to Appendix A.2 for details.

NER Baselines We compare CrossAgentIE against the following baselines: (1) **Self-consistency** (Wang et al., 2023d), which aggregates multiple outputs via voting to improve stability. (2) **Soft Self-consistency** (Wang et al., 2024), which softens voting decisions using uncertainty-aware aggregation. (3) **All-Entity-in-One (AEiO)** (Li et al., 2024a), which extracts multiple entity types in a single model, handling all categories together (e.g., "Identify person, location, and organization entities in the sentence"). (4) **Type-Agents**, which uses multiple specialized LLM prompts, each focused on a specific type. (5) **Template fine-tuning**, which fine-tunes a single LLM using distilled outputs.

RE Datasets We evaluate RE performance on CONLL2004, Semeval, TACRED, NYT (Face, 2025), and SciERC (Luan et al., 2018). Please refer to Appendix A.2 for details.

RE Baselines We compare CrossAgentIE against the following baselines: (1) **One-step** (Li et al., 2024a), which jointly extracts entities and their relations within a single prompt in a structured format, (2) **Direct-prompting**, which extracts relation triplets in a single step. (3) **Type-Agents** and (4) **Template fine-tuning**, which follow the same configurations as in the NER.

Implementation Details We conduct zero-shot experiments using GPT-3.5-Turbo (OpenAI). Each entity type is assigned a dedicated type agent, ensuring one-to-one mapping with the entity label set. Our framework is built on Microsoft's open-source Autogen [1]. We set the temperature to 0.9, the maximum number of iterations to 2, and the frequency penalty to 0.1.

Metrics and Evaluation We compute micro-averaged precision, recall, and F1-score [2] using a strict span-level matching, where only exact matches with ground truth entities count as true positives. See Appendix A.3 for details.

4.2 Main Results

We evaluate the performance of all methods using micro F1-scores across NER and RE test sets.

Main Results in NER As our main NER results, Table 1 presents the F1-scores achieved by GPT-3.5 using various prompting strategies. The effectiveness of CrossAgentIE is evident, as it consistently outperforms both the AEiO approach and Type-Agents across all datasets, achieving an average F-1 score improvement of 8.56% over AEiO and 7.63% over Type-Agents. We further compare CrossAgentIE with the existing Self-consistency (SC) and Soft Self-consistency framework to validate the effectiveness. As shown in Table 4, CrossAgentIE outperforms SC and Soft SC by 5.97% and 11.32% in F1 score, respectively.

Main Results in RE As our main RE results, Table 5 presents the F1 scores achieved by GPT-3.5 across different methods. Compared to Direct-Prompting and Type-Agents, CrossAgentIE achieves an average F1 improvement of 9.13% over Direct-Prompting, and 5.54% over Type-Agents across all datasets, highlighting its robustness in relation extraction.

Results in Template Fine-tuning Table 1 and 5 show that template fine-tuning improves performance over zero-shot inference. On the CONLL04 NER dataset, the AEiO method achieves an F1-score of 58.13%, while template fine-tuning boosts it to 68.38%, a 10.25% increase. Across all datasets, the template fine-tuned GPT-3.5 outperforms all baselines, improving NER performance over AEiO by an average of 7.25% and RE performance over Direct-Prompting by 6.08%.

[1] https://microsoft.github.io/autogen/

[2] https://scikit-learn.org/stable/index.html

Method	**CONLL03**	**CONLL04**	**SemEval**	**TACRED**	**OntoNotes**	**Average**
AEiO (Li et al., 2024a)	49.65	58.13	30.10	30.79	39.47	41.63
CrossAgentIE						
- Type-Agents	64.65	55.73	29.28	28.47	37.69	43.16
- CrossAgentIE	**72.94**	$\underline{66.45}^{\dagger}$	**33.87**	**32.52**	**45.18**	**50.19**
- Template-finetuning (One-LLM)	$\underline{71.78}^{\dagger}$	**68.38**	$\underline{31.17}^{\dagger}$	$\underline{31.49}^{\dagger}$	$\underline{41.56}^{\dagger}$	$\underline{48.88}^{\dagger}$

Table 1: The micro F1 scores (%) of GPT-3.5 on the NER datasets with different prompting strategies. † indicates the suboptimal performance.

Method	**F1**
G&O (GPT-3.5 only) (Li et al., 2024a)	63.94
-One-step	44.77
- AEiO	49.65
Self-Improving(Xie et al., 2024)	
- Naive zero-shot prompting	68.97
- Entity-level threshold filtering	**74.99**
- Sample-level threshold filtering	73.97
- Two-stage majority voting	74.51
CrossAgentIE	
-Type-Agents	64.65
-CrossAgentIE	72.94
-Template-finetuning (One-LLM)	71.78

Table 2: NER results (%) on CONLL03. Bold numbers represent the highest score for zero-shot approaches.

Method	**CONLL04**
G&O (Li et al., 2024a)	33.50
-One-step	38.70
CrossAgentIE	
-Direct-prompting	33.59
-Type-Agents	30.91
- CrossAgentIE	**40.06**
-Template-finetuning (One-LLM)	35.18

Table 3: F1 scores (%) of GPT-3.5 on the RE task—CONLL04 using different strategies.

Fairness and Bias Control in Debate To ensure fairness, all type agents have equal weights, preventing any single agent from dominating classification. The speaking order is randomized to eliminate positional bias. If no consensus is reached, the Summarizer LLM aggregates evidence and confidence scores for the final decision, as detailed in Section 3.3. These mechanisms ensure an unbiased and balanced debate.

Additional Results We evaluate the self-verification reasoning (Weng et al., 2023) within the Type-Agents baseline across various backbone models. As shown in Figure 3, self-verification drags down the performance in zero-shot settings across datasets and model sizes despite its complexity. See Appendix C for further analysis.

Method	**CONLL04**
Self-consistency (SC) (Wang et al., 2023d)	60.48
Soft SC (Wang et al., 2024)	55.13
CrossAgentIE	
-Type-Agents	55.73
- CrossAgentIE	$\underline{66.45}^{\dagger}$
-Template-finetuning (One-LLM)	**68.38**

Table 4: F1 scores (%) of GPT-3.5 on the NER task—CONLL04 using different strategies.

4.3 Ablation Studies

To evaluate the contribution of key components in our approach, we conduct ablation studies focusing on five aspects: (1) comparison with other zero-shot methods (2) backbone model selection (3) model structure design (4) effectiveness of conflict debate and (5) template fine-tuning optimization. These studies quantify the impact of each component on both NER and RE.

NER Baselines Comparison We compare our approach with existing zero-shot LLM methods for NER, including G&O (Li et al., 2024a), a simple but effective work to analyze the GPT-3.5's zero-shot performance on IE tasks; Self-Improving for Zero-Shot NER with LLM (Xie et al., 2024), which enhances zero-shot NER through self-annotation, pseudo-demonstrations, and consistency-based filtering; and Decomposed-QA (Xie et al., 2023b), which explores zero-shot NER with ChatGPT. As shown in Table 2 and 6, CrossAgentIE outperforms G&O by 7.07% and Self-Improving by 0.56% in F1 score on the CoNLL03, while surpassing Decomposed-QA by 5.98% F1 score on the OntoNotes. Furthermore, under the zero-shot setting with a single LLM, our template fine-tuned model exceeded G&O by 5.91% and Decomposed

Method	CONLL04	TACRED	SemEval	NYT	SCIREC	Average
One-Step (Li et al., 2024a)	38.70†	39.27	15.03	10.55	11.71	23.05
Direct-prompting	33.59	32.51	17.50	10.97	14.65	21.84
CROSSAGENTIE						
- Type-Agents	30.91	43.93	19.48	14.06	18.76	25.43
- CROSSAGENTIE	**40.06**	**46.81**	**23.08**	19.18†	**23.73**	**30.97**
- Template-finetuning (One-LLM)	35.18	42.54†	20.69†	**21.62**	19.57†	27.92†

Table 5: The micro F1 scores(%) of GPT-3.5 on the RE datasets with different prompting strategies.† indicates the suboptimal performance.

Method	F1
Decomposed-QA (Xie et al., 2023b)	37.45
Vanilla	33.74
Syntactic prompting	39.00
Tool augmentation	39.20
CROSSAGENTIE	
-Type-Agents	37.69
-CROSSAGENTIE	**45.18**
-Template-finetuning (One-LLM)	41.56

Table 6: NER results (%) on OntoNotes. Bold numbers represent the highest score for zero-shot approaches.

CROSSAGENTIE	F1
NER	
-Type-Agents	68.61
-CROSSAGENTIE	72.14
-Template-finetuning (One-LLM)	70.69
RE	
-Type-Agents	49.79
-CROSSAGENTIE-RE	55.22
-Template-finetuning (One-LLM)	40.67

Table 7: Performance(%) on CONLL04 with GPT-4o.

by 2.36%, further demonstrating its effectiveness.

RE Baselines Comparison We compare our approach with existing zero-shot LLM methods on RE task, including One-step and G&O (Li et al., 2024a). As shown in Table 3, CROSSAGENTIE outperforms One-step by 5.63% and G&O by 10.83% in F1 score on the CoNLL04 dataset. Under the zero-shot setting with a single LLM, our template fine-tuned model surpasses One-step by 2.48% and G&O by 7.68%.

Backbone Model Selection Our experiments utilize GPT-3.5[3], LlaMa3-8b[4], Mistral-7B (Jiang et al., 2023) and Mixtral 8x7B (Jiang et al., 2024a) as backbone LLMs. Figure 3 presents their NER performance across three evaluation settings: Type-Agents, Self-Verification, and Our method. Regardless of the reasoning method used, GPT-3.5 consistently outperforms the other models in precision, recall, and F1-score, highlighting the significant impact of a stronger backbone model on overall performance. This reinforces GPT-3.5 as the optimal choice for our debate-driven multi-agent framework. Additionally, we evaluate our approach using GPT-4o[5], with results on the CoNLL04 dataset presented in Table 7. For a detailed comparison of Type-Agents NER baselines and additional details, please refer to Appendix A.1.

Framework Design Comparison While a strong backbone model is essential, the reasoning framework is equally crucial. A single-step summarization approach reduces computational costs by summarizing first-round responses instead of iterative reasoning. However, this sacrifices refinement and deeper reasoning, key strengths of our debate-driven framework. To evaluate this trade-off, we compared both methods, with results in Appendix I confirming our structured debate's superior performance and efficiency.

Conflict Resolution Efficiency Entity classification conflicts pose a key challenge in our multi-agent debate system. We analyzed 300 CoNLL03 documents, identifying 688 conflict instances, of which 77.5% are successfully resolved in a single debate turn. Among the unresolved cases, 35 are false positives, and only 6 require additional rounds, demonstrating the system's efficiency in handling complex cases.

Effectiveness of Structured Debate We assess the impact of structured debate on NER and RE through an ablation study on CoNLL04, comparing four configurations: (1) Type-Agents without de-

[3] https://platform.openai.com/docs/models#gpt-3.5

[4] https://ai.meta.com/blog/meta-llama-3/

[5] https://platform.openai.com/docs/models#gpt-4o

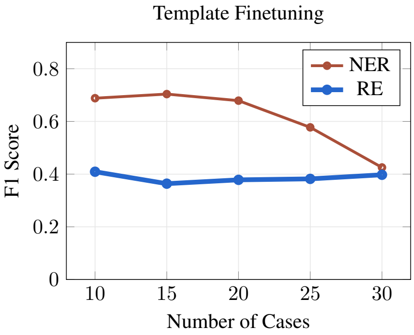

Figure 2: Template Fine-tuning Cases on CONLL04

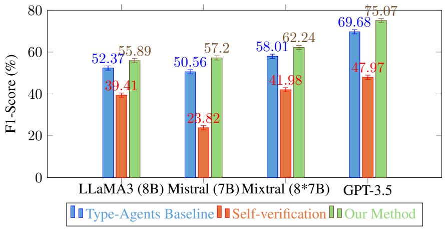

Figure 3: Performance (%) of different LLMs of NER on CONLL03.

bate, (2) Debate for RE only, (3) Debate for NER only, and (4) Debate for both. Table 13 shows that structured debate enhances performance by refining entity classification and resolving label ambiguities, as detailed in Appendix G. To further assess the benefits of iterative NER-RE interactions, we conduct a second-round feedback experiment, where cross-task refinements improve predictions. As shown in Table 12, this iterative feedback boosts recall by recovering missed entities and refining relation classification. See Appendix H for details.

Template Fine-tuning Optimization Our template fine-tuning mechanism aims to match the performance of multi-agent refinement. To optimize a single LLM for maximum accuracy, we explore the optimal number of cases needed to achieve the best F1-score. By varying the case count in NER and RE tasks on the CONLL04 dataset (Figure 2), we find that the optimal number is 5 cases per type for NER and 3-4 cases per type for RE. Please see Appendix J for more details.

Cost and Time Efficiency We evaluate cost per data point and time consumption for long and short debates. Using the *Efficiency Score* as a measure of cost-effectiveness, our framework optimally balances computational efficiency and performance. The final results depend on the required debate rounds per dataset, demonstrating its practicality for scalable applications. Please see Appendix L.

Error Types	**Baseline-NER**	**1st-Debate-NER**	**2nd-Feedback-NER**
Boundary Errors	90	81	90
Wrong types	333	251	343
Missing Entities	686	680	618
Total	**1109**	**1012**	**1051**

Table 8: Error Type Counts on CONLL04 for NER: Comparison of Baseline, 1st-Round Cross-Type Discussion, and 2nd-Round Cross-Task Discussion. Bold numbers indicate total errors, showcasing reductions achieved by our methods.

4.4 Case study

Error Analysis We analyze errors in our multi-agent framework on the CONLL04 dataset, categorizing them into three types to identify model limitations and guide improvements.

Error Types and Statistics Table 8 summarizes error statistics, categorizing errors into wrong type errors, boundary errors, and missing entities. 1) Wrong type errors occur when an entity is assigned an incorrect type from the predefined label set. 2) Boundary errors arise when the predicted span misaligns with the gold annotation, either by fully containing, being contained within, or partially overlapping it. 3) Missing entities refer to undetected gold entities. Additionally, we consider spurious entity errors, where the model predicts non-existent entities, though our primary focus remains on the three main error types. For a detailed breakdown of error distribution, impact across model stages, and case studies, see Appendix K for details.

Case Study of Error Correction and Error Increase As shown in Table 8, Cross-task Debating effectively reduces Boundary Errors and Wrong Types errors. In the Baseline stage, errors are dominated by false negatives (FN) and false positives (FP), leading to suboptimal performance. The 1st-Debate-NER stage significantly reducesFP and slightly decreases FN, improving precision and F1-score. The 2nd-Feedback-NER stage further reduces FN, achieving an 8.73% recall improvement with a minor FP increase. This demonstrates that when FN are the primary source of error, RE-based knowledge augmentation in 2nd-Feedback-NER effectively reduces FN, boosting recall and F1-score. Despite a slight FP increase, the FN reduction leads to net performance gains. Please see Appendix K.

5 Conclusion

In this paper, we propose CROSSAGENTIE, a cross-type and -task multi-agent collaboration framework designed to enhance structured prediction in information extraction (IE) tasks using LLMs. Unlike conventional zero-shot strategies, CROSSAGENTIE introduces two collaboration mechanisms that enable mutual refinement between NER and RE tasks, improving prediction accuracy. Additionally, we develop template fine-tuning to consolidate output knowledge into a single model, significantly enhancing efficiency. Test under zero-shot IE settings with GPT-3.5, our bidirectional collaboration and template fine-tuning achieve substantial performance gains, demonstrating the effectiveness of CROSSAGENTIE. Ablation studies further validate the efficiency of each component in our multi-agent system, while evaluations across diverse LLMs and datasets demonstrate the generalizability of CROSSAGENTIE. We hope our work inspires future research on multi-agent collaboration frameworks in LLMs and contributes to the development of effective and interpretable IE systems.

Acknowledgement

Our work is sponsored by the NSF NAIRR Pilot with PSC Neocortex and NCSA Delta, Commonwealth Cyber Initiative, Children's National Hospital, Fralin Biomedical Research Institute (Virginia Tech), Sanghani Center for AI and Data Analytics (Virginia Tech), Virginia Tech Innovation Campus, and generous gifts from Cisco Research and the Amazon + Virginia Tech Center for Efficient and Robust Machine Learning.

Limitations

Due to computational constraints, our evaluation was conducted on a limited set of datasets and tasks. While these experiments demonstrate the effectiveness of CROSSAGENTIE, incorporating more domain-specific datasets could further enhance the robustness of our conclusions. Below, we outline key limitations of our approach.

Computational Cost Our multi-agent framework incurs additional computational overhead due to iterative debate and bidirectional refinement. Although template fine-tuning reduces inference costs, the initial debate process remains expensive, particularly for large-scale datasets.

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

Dependency on Model Accuracy The framework relies on the reasoning capabilities of LLMs, which can still produce hallucinated or inconsistent outputs. While intra-group and inter-group debates help mitigate errors, misclassifications in entity recognition and relation extraction may still occur. Additionally, due to 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, which limits its adaptability to open-domain or evolving schemas. Extending it to dynamic ontologies would require additional mechanisms for expansion and adaptation.

Ethics

In this work, we propose a method to improve LLM performance on the fundamental task of relation extraction. We do not anticipate any ethical issues regarding the topics of this research.

References

Meta AI. Meta llama 3. https://ai.meta.com/blog/meta-llama-3/. Accessed: 2025-01-12.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *Preprint*, arXiv:2305.10403.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

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.

Young-Min Cho, Raphael Shu, Nilaksh Das, Tamer Alkhouli, Yi-An Lai, Jason Cai, Monica Sunkara, and Yi Zhang. 2024. Roundtable: Investigating group decision-making mechanism in multi-agent collaboration. *Preprint*, arXiv:2411.07161.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

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.

Andrew Estornell and Yang Liu. 2024. Multi-LLM debate: Framework, principals, and interventions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Hugging Face. 2025. Nyt dataset from irds. `https://huggingface.co/datasets/irds/nyt`. Accessed: 2025-01-10.

Chufan Gao, Xulin Fan, Jimeng Sun, and Xuan Wang. 2024a. PromptRE: Weakly-supervised document-level relation extraction via prompting-based data programming. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 132–145, Bangkok, Thailand. Association for Computational Linguistics.

Chufan Gao, Xuan Wang, and Jimeng Sun. 2024b. Ttm-re: Memory-augmented document-level relation extraction. *Preprint*, arXiv:2406.05906.

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

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Guochao Jiang, Ziqin Luo, Yuchen Shi, Dixuan Wang, Jiaqing Liang, and Deqing Yang. 2024b. ToNER: Type-oriented named entity recognition with generative language model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16251–16262, Torino, Italia. ELRA and ICCL.

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. 2023. Instruct and extract: Instruction tuning for on-demand information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore. Association for Computational Linguistics.

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

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent advances in named entity recognition: A comprehensive survey and comparative study. *Preprint*, arXiv:2401.10825.

Seoyeon Kim, Kwangwook Seo, Hyungjoo Chae, Jinyoung Yeo, and Dongha Lee. 2024. Verifiner: Verification-augmented ner via knowledge-grounded reasoning with large language models. *Preprint*, arXiv:2402.18374.

Yinghao Li, Colin Lockard, Prashant Shiralkar, and Chao Zhang. 2023. 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. 2024a. 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.

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

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.

Yantao Liu, Zijun Yao, Xin Lv, Yuchen Fan, Shulin Cao, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Untangle the KNOT: Interweaving conflicting knowledge and reasoning skills in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17186–17204, Torino, Italia. ELRA and ICCL.

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.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. Gpt-3.5 turbo documentation. `https://platform.openai.com/docs/models/gpt-3-5-turbo`. Accessed: 2024-06-15.

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. 2023c. Gpt-4 is openai's most advanced system, producing safer and more useful responses. Accessed: November 10, 2023.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *Preprint*, arXiv:1906.02243.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. *Preprint*, arXiv:2305.02105.

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

Qing Wang, Kang Zhou, Qiao Qiao, Yuepei Li, and Qi Li. 2023a. Improving unsupervised relation extraction by augmenting diverse sentence pairs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12136–12147, Singapore. Association for Computational Linguistics.

Shuhe Wang, Yuxian Meng, Rongbin Ouyang, Jiwei Li, Tianwei Zhang, Lingjuan Lyu, and Guoyin Wang. 2022. Gnn-sl: Sequence labeling based on nearest examples via gnn. *Preprint*, arXiv:2212.02017.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. 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.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023c. Instructuie: Multi-task instruction tuning for unified information extraction. *Preprint*, arXiv:2304.08085.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. 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.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. *Preprint*, arXiv:2212.09561.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *Preprint*, arXiv:2308.08155.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023a. 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.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023b. Empirical study of zero-shot ner with chatgpt. *Preprint*, arXiv:2310.10035.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 583–593, Mexico City, Mexico. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada. Association for Computational Linguistics.

Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, and Alakananda Vempala. 2024a. ULTRA: Unleash LLMs' potential for event argument extraction through hierarchical modeling and pair-wise self-refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8172–8185, Bangkok, Thailand. Association for Computational Linguistics.

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

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, Shiding Zhu, Jiyu Chen, Wentao Zhang, Xiangru Tang, Ningyu Zhang, Huajun Chen, Peng Cui, and Mrinmaya Sachan. 2023. Agents: An open-source framework for autonomous language agents. *Preprint*, arXiv:2309.07870.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. Universalner: Targeted distillation from large language models for open named entity recognition. *Preprint*, arXiv:2308.03279.

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 Detailed Experiment Setup

A.1 Models

Our research focuses on GPT-3.5, specifically the gpt-3.5-turbo (OpenAI, 2023a) [6]. While it is not the latest model, we use it to maintain experimental consistency. For open-source LLMs, we employ Llama 3-8B (AI) [7], Mistral-7B (Jiang et al., 2023) [8], and Mixtral 8x7B (Jiang et al., 2024a) [9]. All experiments involve forward inference only, except for template fine-tuning. GPT-3.5 inference is conducted through the OpenAI API, while open-source models run on HuggingFace Transformers. Llama 3-8B and Mistral-7B are deployed on single NVIDIA A100 80G GPUs, and Mixtral 8x7B runs on two GPUs. Our multi-agent debate framework utilize Microsoft's open-source Autogen (Wu et al., 2023) [10]. For template fine-tuning, we use the gpt-3.5-turbo-1106 version [11], following OpenAI's official fine-tuning guidelines [12]. Details on fine-tuning dataset construction and analysis are provided in Appendix J.

	CONLL03	CONLL04	SemEval	TACRED	OntoNotes
n-instance	3453	288	2717	15509	8262
n-entity-type	4	3	2	17	18
n-entity-mention	4945	844	5434	31018	11257

Table 9: NER dataset statistics.

	CONLL04	TACRED	SemEval	NYT	SCIREC
n-instance	288	446	2717	369	1088
n-entity-type	5	4	10	7	7
n-entity-mention	42	446	2717	265	974

Table 10: RE dataset statistics.

A.2 Datasets

NER In the NER task, we use datasets from multiple sources: CoNLL2003 (Tjong Kim Sang and De Meulder, 2003), CoNLL2004 (Carreras and Màrquez, 2004), OntoNotes4 (Pradhan et al., 2013), TACRED (Zhang et al., 2017), and SemEval2010 (Hendrickx et al., 2010). The CoNLL2003 and CoNLL2004 datasets are sourced from (Li et al., 2024b), while TACRED and SemEval come from the processed versions in (Wan et al., 2023). We preprocess all datasets to align with our study while preserving their original structure. Specifically, we extract labeled phrases from each sentence, group them by entity type, and use them as ground truth for computing the micro F1-score per doc_id. For instance, CoNLL2004 contains three label types (PER, ORG, and LOC), and we exclude "MISC" for fair comparison with G&O. For TacRed, we test our method on six label types to verify our effectiveness (ORG, PER, LOC, Country, City, Nationality, and URL). For GPT-3.5, we process entire paragraphs. In contrast, other LLMs receive sentence-level inputs due to memory constraints. All models are provided with raw sentences without labeled entities. For simplicity, we briefly refer to CoNLL2003 as CoNLL03 and CoNLL2004 as CoNLL04 throughout the paper for consistency. We report the performance on the test set of each dataset, and the detailed statistics are shown in Table 9.

RE For the RE task, we use CoNLL2004 (Carreras and Màrquez, 2004), NYT (Face, 2025), SemEval 2010 (Hendrickx et al., 2010), TACRED (Zhang et al., 2017)and SciERC (Luan et al., 2018). Consistent with NER, NYT is sourced from (Wang et al., 2023c) and SciERC from (Wan et al., 2023). For TACRED, we retain only four relation types to evaluate the effectiveness of our framework: "organization has member", "organization has website", "per: cities_of_residence" and "person_has_age". In SemEval 2010, subjects and objects are treated as independent agents to align with our workflow. When type-specific agents generate no conflicts, we skip the debate stage and proceed directly to bidirectional refinement and template fine-tuning. To improve agent understanding, we provide natural language explanations for relation labels. For example,"per: cities_of_residence" is defined as "a person lives or has lived in a city as their place of residence". We report the performance on the test set of each dataset, and the detailed statistics are shown in Table 10.

A.3 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

[6] platform.openai.com/docs/models/gpt-3-5-turbo

[7] https://huggingface.co/unsloth/Meta-Llama-3.1-8B-bnb-4bit

[8] huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[9] huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

[10] https://microsoft.github.io/autogen/

[11] https://platform.openai.com/docs/models

[12] https://platform.openai.com/docs/guides/fine-tuning

type, the corresponding list remains NULL. For RE, we extract head-tail entity pairs for each relation type, leaving the output empty when no valid pairs exist.

Due to their generative nature, LLMs often introduce noise during post-processing, 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.

We filter noisy content by matching generated outputs with original sentences to mitigate these issues. 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.

We provide natural language explanations that explicitly define the expected entity types for each relation to maintain the correct logical order between the head and tail entities. 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 template-based 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. For OntoNotes, we first filter the data points based on type labels to extract those containing the target entity types. Then, we use type-specific agents to identify the representative entities for each type. During the fine-tuning phase, we use a single model to perform entity recognition for all types across 100 data points. We randomly sampled 100 data points from each dataset's test set for method evaluation.

B Baseline selection

This section categorizes and introduces key research on LLM-based NER and RE, highlighting approaches distinct from our setting.

LLMs for NER Beyond our zero-shot setting, LLM-based NER methods generally follow two paradigms: few-shot/in-context learning and supervised fine-tuning. Few-shot approaches primarily leverage in-context learning (ICL), providing labeled examples within prompts to guide predictions. For example, GPT-NER (Wang et al., 2023b) frames NER as a text generation task, employing entity markers and self-verification to mitigate over-predictions. ProGen (Heng et al., 2024) enhances this paradigm with few-shot learning through step-by-step generation and self-reflection, improving dataset quality rather than directly extracting entities. Supervised fine-tuning methods explicitly train models on annotated or synthetic datasets. For example, UniversalNER (Zhou et al., 2024) employs instruction tuning and targeted distillation to train an LLaMA-based model, leveraging ChatGPT-generated synthetic data for cost-efficiency and domain generalization. VerifiNER (Kim et al., 2024) focuses on post-hoc verification, utilizing external knowledge bases to refine entity boundaries and classifications.

LLMs for RE Beyond our zero-shot setting, LLM-based RE methods follow two main paradigms: few-shot in-context learning and supervised fine-tuning. Few-shot approaches extract relational information without fine-tuning. For example, GPT-RE (Wan et al., 2023) enhances in-context learning by optimizing example retrieval and incorporating reasoning-based augmentation, improving alignment between input text and relation labels. Supervised fine-tuning explicitly trains models for RE. For example, URE (Wang et al., 2023a) refines relational embeddings through contrastive learning and margin loss within a BERT-based framework. QA4RE (Zhang et al., 2023) reframes RE as a multiple-choice QA task, aligning LLM predictions with structured relation templates using instruction-tuned datasets.

Nonetheless, existing studies have overlooked the challenges of LLMs' performance in structured prediction with mixed prompts and have yet to fully explore their embedding-level capabilities for enhancing NER and RE performance, which are the central topics of our research.

C More Results Analysis

Additional Analysis Table 11 summarizes the existing methods, including supervised fine-tuning, few-shot learning, and in-context learning, and their results for NER on CONLL03 and RE on TACRED. Although our framework falls behind advanced tuning-based methods, the performance gap has narrowed. Three key factors drive these improvements over zero-shot baselines: 1) The multi-agent debate enables dynamic collaboration among agents and allows iterative refinement of entity and relation predictions. 2) Ontology-guided learning leverages structured ontology information to enhance agents' comprehension of NER and RE, providing a systematic framework for entity categorization and relation modeling. 3) Enriched knowledge integration incorporates task-specific contextual information, offering richer semantic cues that improve prediction accuracy. We further analyze the effectiveness of structured debate components in Appendix G.

D Detail prompts for NER

Type-Agent Prompt Below is an example of the prompt design for the Type Agent in the NER task:

Listing: PER_Agent

```
You are a knowledgeable assistant who
    specializes in recognizing and
    understanding named entities.
<Human>Given the following text, extract
     all the 'Person' named entities and
     return the result in the following
    format:
    <bot> Response: ###list of extracted
         persons and confidence scores
        ###.
    Include "###" before and after each
        extracted entity and confidence
        score.
    Person entities are named persons or
         families. For each extracted
        entity, assign a confidence
        score between 0 and 1 based on
        how certain you are about the
        entity's classification.
    Return the extracted entities along
        with their confidence scores in
        the specified format.
    Text: {text}
    <bot> Response:
```

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.

We adopt the All-Entity-in-One (AEiO) approach from G&O (Li et al., 2024a) as our baseline, a method that generates all entities simultaneously, as shown below. The AEiO approach performs both information extraction and structuring in a single step.

Cross-Type prompt When conducting cross-type debates to resolve conflicts, we first identify conflicts where multiple entity labels are assigned to the same entity within a sentence, as shown below.

Example of Cross-Type Conflicts-NER

```
{
    "doc_id": "1",
    "sentence": "An art exhibit at the
        Hakawati Theatre in Arab east
        Jerusalem was a series of
        portraits of Palestinians killed
         in the rebellion.",
    "entity": "Hakawati Theatre",
    "conflict_types": [
        "LOC",
        "ORG"
    ]
},
{
    "doc_id": "2",
    "sentence": "PERUGIA , Italy ( AP )"
        ,
    "entity": "PERUGIA",
    "conflict_types": [
        "LOC",
        "ORG"
    ]
},
{
    "doc_id": "3",
```

Method	Ontology Usage	Paradigm	CONLL03	TACRED
GPT-NER (Wang et al., 2023b)	✗	SFT	89.97	-
GNN-SL (Wang et al., 2022)	✗	SFT	93.20	-
GPT-RE_FT (Wan et al., 2023)	✗	SFT, FCL-15	-	72.14
O&G- GPT3.5 only(Li et al., 2024b)	✓	ZS	63.94	-
Self-improving_ZS (Xie et al., 2024)	✓	ZS	**74.51**	-
Self-improving_Demo (Xie et al., 2024)	✗	ICL-full	83.51	-
GPT-RE_SimCSE (Wan et al., 2023)	✗	FCL-15	-	37.44
QA4RE (Zhang et al., 2023)	✓	ZS	-	44.2†
Debate-NER (GPT-3.5)	✓	ZS	72.94†	-
Debate-RE (GPT-3.5)	✓	ZS	-	**46.81**

Table 11: NER results (%) on CONLL03 and RE results on TACRED. Bold numbers represent the highest score for zero-shot approaches.† represents the second-best. SFT denotes supervised fine-tuning. FCL denotes few-shot learning. ICL denotes in-context learning, and ICL-Full denotes with the full training dataset.

```
    "sentence": "Reagan sounded positive
         notes reminiscent of earlier
        speeches throughout his
        political career _ the pre-
        eminent position of ‘ ‘ We the
        People ’ ’ in the American
        system , the image of America as
         a shining ‘ ‘ city upon a hill
        , ’ ’ the importance of paying
        more attention to American
        history.",
    "entity": "America",
    "conflict_types": [
        "LOC",
        "ORG"
    ]
},
{
    "doc_id": "3",
    "sentence": "Reagan sounded positive
         notes reminiscent of earlier
        speeches throughout his
        political career _ the pre-
        eminent position of ‘ ‘ We the
        People ’ ’ in the American
        system , the image of America as
         a shining ‘ ‘ city upon a hill
        , ’ ’ the importance of paying
        more attention to American
        history.",
    "entity": "Regan",
    "conflict_types": [
        "LOC",
        "PER"
    ]
}
```

Example of Cross-Type Conflicts-RE

```
    {
        "doc_id": "11",
        "entity": [
            "MILAN",
            "Italy"
        ],
        "conflict_types": [
            "Organization-based-in",
            "Located-in"
        ],
        "sentence": "MILAN , Italy ( AP
            )"
    }
```

Next, we use the following prompts to construct the conflict resolution discussion framework. Similar to the design of Type agents, the prompts for the debate framework follow the approach illustrated in Listings 5-6.

List-5: Person_agent

```
system_message = "You determine if the
    entity belongs to a person.",
description = "Responsible for
    determining if an entity is a person
     or people. For each determination,
    assign a confidence score between 0
    and 1 based on how certain you are
    about the classification.",
confidence = "The confidence score
    reflects the certainty of the agent
    in classifying the entity as a
    person."
```

List-6: Location_agent

```
system_message = "You are a specialized
    agent responsible for verifying if
    an entity belongs to the Location
    type.",
description = "Responsible for
    determining if an entity is a
    location, which includes politically
     or geographically defined locations
     such as cities, provinces,
    countries, international regions,
    bodies of water, mountains, etc. For
     each determination, assign a
    confidence score between 0 and 1
    based on how certain you are about
    the classification.",
confidence = "The confidence score
    reflects the certainty of the agent
    in classifying the entity as a
    location."
```

List-7: Organization_agent

```
system_message = "You are a specialized
    agent responsible for verifying if
    an entity belongs to the
    Organization type.",
description = "Responsible for
    determining if an entity is an
    organization, which includes named
    corporate, governmental, or other
    organizational entities. For each
    determination, assign a confidence
    score between 0 and 1 based on how
    certain you are about the
    classification.",
confidence = "The confidence score
    reflects the certainty of the agent
    in classifying the entity as an
    organization."
```

The prompt to initiate group debate:
Conflict Resolution Group_chat Debate

```
chat_result = initiator_agent.
    initiate_chat(
     group_chat_manager,
     message=(
         f"The entity '{entity}' appears
             in the context: '{sentence}
             '. "
         f"There is a conflict between {
             Location} agent and {
             Organization} agent over
             which type this entity
             belongs to. "
         f"The {Location} agent has
             assigned a confidence score
             of {location_confidence} to
             classify the entity as '
             Location', "
         f"while the {Organization} agent
              has assigned a confidence
             score of {
             organization_confidence} to
             classify the entity as '
             Organization'. "
         f"Based on the given context and
              confidence scores, please
             discuss and decide which
             type the entity '{entity}'
             should belong to."
    ),
)
```

Each Type Agent resolves conflicts by generating a new response based on a conflict-specific prompt, leveraging sentence context and confidence scores to refine its reasoning. These prompts guide agents in justifying their predictions, providing confidence levels, and considering arguments from conflicting agents.

The structured validation process requires agents to critically assess evidence, including contextual cues, boundary definitions, label-specific characteristics, and confidence scores. The final label is assigned based on logical reasoning, contextual alignment, and confidence level comparison. If consensus is reached, the agreed label is assigned. When confidence scores vary significantly, the agent with the highest score prevails. If no consensus is achieved, unresolved conflicts are escalated for further analysis or external review.

This process is particularly relevant when multiple Type Agents classify the same entity under different labels, such as both Person and Organization agents claiming the same entity. By integrating confidence scores and iteratively resolving conflicts, the Cross-Type Debate Process enhances classification precision, ensuring accurate labeling with minimal ambiguity.

E Detail prompts for RE

RE is more challenging than NER as it requires not only entity identification but also contextual relationship interpretation. Ambiguous relation labels, such as “place lived” or “located in,” often confuse LLMs. To mitigate this, we take a two-step approach: first, we design tailored prompts to improve contextual understanding; second, we use relation logic to define type constraints for head and tail entities, reinforcing their semantic roles.

For the RE task, Listings 8–9 illustrate how to construct a Relation Type Agent using examples from two relation types.

List-8: Killer_Victim_Relationship

```
% Please identify the "Killer kills the
    Victim" relationship in the
    paragraph,
% which means a person (Killer) causes
    the death of another person (Victim)
    .
% This relationship is often expressed
    in the form of "Killer kills the
    Victim".
% Use the provided candidate entities as
     a reference, but also recognize
% any other entities in the sentence if
    necessary.

- Sentence: "{sentence}"
- Candidate Entities: {entities}
- Task: Identify all pairs of entities
    involved in a "Kill" relationship.

<bot> Response: ["Head": "###entity###",
     "Tail": "@@@entity@@@"]
% Include "###" to identify the Head
    entity and "@@@" to identify the
    Tail entity.

% Return the identified pairs of
    entities in this specified format,
% ensuring clarity and accuracy.
```

List-9: Person_Location_Relationship

```
% Please analyze the given paragraph to
    identify any instances where it
    implies or states
% that a person resides or has resided
    in a specific location.
% This relationship is between a person
    and a location, where the person has
     lived in the location.
% The person is the head entity and the
    location is the tail entity.
% Use the provided candidate entities as
     a reference, but also consider any
    other entities
% in the sentence if necessary.

- Sentence: "{sentence}"
- Candidate Person Entities: {
    person_entities}
- Candidate Location Entities: {
    location_entities}
- Task: Identify all pairs of entities
    where a person resides or has
    resided in a location.

% Format your response as follows:
% - Head Entity (Person): ###entity###
% - Tail Entity (Location): @@@entity@@@

% Example:
<bot> Response: ["Head": "###John Smith
    ###", "Tail": "@@@New York@@@"]

% Return the identified pairs of
    entities in this specified format,
% ensuring clarity and accuracy.
```

Furthermore, we use the One-Step RE prompt, adapted from G&O (Li et al., 2024a), as our baseline, simplifying the process into a single prompt.

Another baseline, Direct Prompting, extracts relational triplets (head, relation, tail) directly from text without explicit entity span classification. This approach prompts a single LLM to identify all relation types in a given sentence and extract head-tail pairs in a single step, while enforcing a predefined output format.

F Mathematical Formulation of Cross-Task Discussion

To better understand the structured interaction between named entity recognition (NER) and relation extraction (RE), we define a complete round of cross-task collaboration. In this process, NER-extracted entities serve as candidates for RE (**NER → RE**), while relational knowledge from RE provides structured feedback to refine entity classification (**RE → NER**). This iterative exchange establishes structured constraints, ensuring consistency between entity extraction and relation identification while maintaining a zero-shot setting.

However, due to the independent nature of NER and RE in zero-shot scenarios, discrepancies often arise between the entity sets used in each task. These inconsistencies introduce a symmetric difference between NER-extracted entities and RE-required entities, leading to additional entity predictions that do not belong to the original entity set of each task. To resolve these inconsistencies, we introduce a cross-task debate mechanism, where NER and RE agents iteratively refine their predictions by minimizing this symmetric difference in their generated entity sets.

The following section presents a formal mathematical formulation of this debate process, detailing how NER and RE collaborate through structured constraints to enforce entity-relation consistency.

NER → RE: Entity Candidates Augmentation. NER agents generate a set of candidate entities $E_{\text{NER}} = \{e_1, e_2, ..., e_k\}$, $e_i = (t_i, l_i, c_{\text{NER}}(e_i))$ where t_i is the extracted entity span, l_i is the predicted entity label, and $c_{\text{NER}}(e_i)$ represents the confidence score. These extracted entities serve as input for RE agents, which predict the relation set: $R_{\text{RE}} = \{(e_p, r, e_q, c_{\text{RE}}(r))\}$ where e_p and e_q are entity pairs, r is the predicted relation, and $c_{\text{RE}}(r)$ is the confidence score. Since NER operates in a zero-shot setting, discrepancies may arise between the extracted entities E_{NER} and those required by (E_{RE}). We define this entity discrepancy as:

$$E_{\Delta} = E_{\text{NER}} \Delta E_{\text{RE}}$$

where $E_{\text{NER}} \setminus E_{\text{RE}}$ represents **spurious entities** extracted by NER but unnecessary for RE, and $E_{\text{RE}} \setminus E_{\text{NER}}$ represents **missing entities** that required by RE but not recognized by NER. To address these inconsistencies, RE agents enforce logical constraints, including hard constraints and soft constraints, to filter out implausible relations and maintain consistency in entity-relation pairs. Hard constraints enforce strict predefined rules by rejecting relations that violate logical structures; for instance, a "Work-for" relation cannot link a Person and a Location, as this contradicts established entity-role mappings. Complementing this, soft constraints incorporate probabilistic rules that guide relation plausibility, aligning predictions with real-world tendencies. For example, organizations are more likely to be headquartered in locations rather than in other entities like persons.

2nd round	**Precision**	**Recall**	**F1**	**Baseline-NER**	**Baseline-RE**	**1st round Debate-NER**	**1st round Debate-RE**	**Direct-RE**
Flow RE–> NER	58.20%	82.49%	68.25%	5.77%		1.80%		
Flow NER–> RE	57.14%	49.14%	52.84%		16.93%		8.51%	19.25%
(+) Self-verification-RE	58.24%	52.09%	54.99%		19.08%		10.66%	21.40%

Table 12: Performance Improvements through 2nd Round Iterative Feedback between NER and RE

By integrating hard constraints (to eliminate invalid relations) and soft constraints (to refine plausible ones), RE agents enhance relational prediction robustness, ensuring alignment with domain knowledge.

RE → NER: Knowledge-base enhancement. After relation extraction, RE agents generate structured knowledge in natural language statements, such as *"John lives in New York"*. These statements are appended to the original input, providing additional contextual signals for NER agents to reassess their classifications. The updated entity set is defined as:

$$E_{\text{updated}} = E_{\text{NER}} \cup (E_{\text{RE}} \setminus E_{\text{NER}})$$

where: $E_{\text{RE}} \setminus E_{\text{NER}}$ represents **new entities** inferred from relational knowledge, and $E_{\text{NER}} \setminus E_{\text{RE}}$ represents **spurious entities** that remain unchanged due to zero-shot constraints. If inconsistencies arise (e.g., an entity previously classified as ORG appears in a "Live-in" relation), a conflict resolution protocol is applied: 1). Conflict Detection: Identify entities whose labels contradict the relational knowledge introduced by RE. 2). Constraint-Based Re-Evaluation: NER agents reassess these entities based on the entity types appearing in the newly introduced relation statements. 3). Final Update: Each NER agent updates its extracted entities and classifications according to the relational context, ensuring alignment with the structured knowledge provided by RE.

To further enhance the reliability of the debate process, our framework integrates external knowledge sources to guide entity classification and relation extraction. A domain ontology provides a structured hierarchy of entity types and their relationships, ensuring classification consistency. For example, "Country" is categorized as a subclass of "Location", enabling a structured classification scheme. In addition to ontology-based guidance, logical constraints enforce consistency and prevent implausible entity-relation assignments. These constraints fall into two categories: Hard constraints, which impose strict rules that must always be satisfied. For instance, a "Person" entity cannot be classified as a "Location", a "Born-in" relation must link a "Person" and a "Location", and a "Work-for" relation cannot exist between two "Location" entities. Soft constraints, which introduce probabilistic guidelines to shape relation plausibility. For example, organizations are more likely to be headquartered in locations rather than in other entity types, and people are more commonly associated with multiple locations over time. By integrating domain ontology and logical constraints, our framework reinforces valid entity-relation structures, enhances model robustness, and ensures adaptability within a zero-shot setting.

		Precision	**Recall**	**F1**
w/o Debate NER 62.48(54.32/73.54)	w/o Debate RE	36.46	35.38	35.91
	Debate RE	47.29	40.79	43.80
Debate NER 66.45(60.45/73.76)	w/o Debate RE	38.44	36.36	37.37
	Debate RE	47.86	41.28	44.33

Table 13: Performance (%) comparison of Baseline and Debate-based NER and RE configurations on CoNLL2004. The results for NER are reported in the format "F1 (Precision / Recall)". w/o Debate represents Type-Agents baseline without debating.

G Effectiveness of Structured Debate

From the results in Table 13, we can draw the following conclusions: (1)Using baseline models for both NER and RE improves performance by 2.32%, demonstrating the benefits of structured integration. (2)Adding the debate mechanism to RE improves performance by 7.89%, effectively resolving ambiguities and enhancing classification. (3) Applying the debate mechanism to NER improves precision and outperforms the baseline by 3.97%, resolving label conflicts. (4) Combining debate-based NER with baseline RE yields a 1.46% improvement by reducing error propagation. These findings confirm the effectiveness of the debate mechanism in addressing challenges collaboratively and enhancing NER and RE performance.

H Enhancing Performance via Second-Round Feedback

To evaluate the impact of iterative interactions between NER and RE, we conducted a second-round feedback experiment on the CONLL04 dataset.

This experiment explores how sequentially leveraging the output of one task (e.g., RE) to refine the other (e.g., NER), and vice versa, enhances predictions. The results, summarized in Table 12,highlight the effectiveness of our iterative mechanism and its contributions to overall performance. From the table, Key observations include: (1) Second-Round Feedback from RE to Improve NER: Compared to baseline NER, integrating RE feedback leads to a 5.77% improvement. Incorporating first-round debate mechanisms further enhances performance by 1.80%, demonstrating the iterative process's role in refining NER predictions based on RE. (2)Second-Round Feedback from NER to Improve RE: Using NER outputs to improve RE in the second round achieves 57.14% Precision, 49.14% Recall, and 52.84% F1, marking a 16.93% gain over baseline RE and an additional 8.51% improvement over first-round debate RE. These results emphasize the mutual reinforcement between NER and RE through circle-based feedback. 3) Incorporating Self-Verification for RE: Adding self-verification to RE results in a total improvement of 19.08% over baseline RE, which is an additional 2.15% gain beyond the 16.93% improvement achieved through second-round feedback from NER. This highlights the role of self-verification in further reducing errors and enhancing RE robustness. By leveraging outputs iteratively, the model resolves ambiguities and reduces error propagation, as evidenced by the substantial improvements across Precision, Recall, and F1 in both tasks. These findings confirm the importance of iterative circle-based mechanisms combined with self-verification in improving the collaborative performance of NER and RE on the CONLL04 dataset.

From the results, we draw the following conclusions: (1)Using baseline models for both NER and RE improves performance by 2.32%, demonstrating the benefits of structured integration. (2)Adding the debate mechanism to RE improves performance by 7.89%, effectively resolving ambiguities and enhancing classification. (3) Applying the debate mechanism to NER improves precision and outperforms the baseline by 3.97%, resolving label conflicts. (4) Combining debate-based NER with baseline RE yields a 1.46% improvement by reducing error propagation. These findings confirm the effectiveness of the debate mechanism in addressing challenges collaboratively and enhancing NER and RE performance.

Second Round Iterative Feedback. To assess the impact of iterative NER-RE interactions, we conducted a second-round feedback experiment on the CONLL04 dataset, refining predictions for both tasks. Results in Table 12 show that additional NER-RE interactions further improve performance for both tasks. Please refer to Appendix H for more details.

I Effectiveness of Summarizer Agent

To explore the impact of CROSSAGENTIE framework designs, we analyze the performance of a system that relies solely on a summarizer. Without effective iterative debates, multi-round summarizer-based interactions fail to ensure consistent improvements. In contrast, our framework—incorporating type-specific agents, debate-driven resolution, and cross-task collaboration—reliably enhances NER and RE precision and recall. Experimental results on CONLL03 (Table 14) show that adding the Summarizer Agent (GPT-3.5) increases recall to 73.51% but lowers precision to 71.04%, resulting in an F1-score of 72.25%. While the summarizer captures broader context, it sacrifices precision due to noise. Further incorporating a two-round discussion with the summarizer and type-specific agents results in precision of 73.02%, recall of 57.05%, and F1 of 64.05%, a notable decline in recall and F1 compared to both the baseline and single-round summarizer. These findings highlight the limitations of summarizer-based multi-round setups and underscore the importance of structured task-specific interactions, such as type-agent debates, in achieving optimal performance for NER and RE.

J Template Fine-tuning

For fine-tuning dataset construction, we follow the guidelines provided by OpenAI's official website. We designed template fine-tuning with the ultimate goal of improving the overall zero-shot IE performance of a single LLM, thereby enhancing efficiency. To determine the optimal number of cases for achieving the best performance, we conducted template fine-tuning experiments on the CONLL04 dataset. The dataset includes three NER entity types: LOC, PER, and ORG, and five RE relation types: Kill, Live-in, Located-in, Organization-based-in, and Work-for.

Case selection. To construct the fine-tuning dataset, we employ an LLM-based selection mechanism. Instead of directly using model-generated outputs, we prompt the LLM to re-evaluate each input-output pair and assign a confidence score to its correctness. These confidence scores are then used to rank the cases in descending order, selecting the highest-ranked ones for fine-tuning. This approach ensures that fine-tuning is guided by the most reliable examples while reducing the inclusion of uncertain or noisy predictions. Confidence scores are derived from the GPT-3.5 model's re-evaluation process.

NER Task Optimization. For NER task, we tested case numbers of 10, 15, 20, 25, and 30. The results indicate that performance peaks at 15 cases before gradually declining as the number of cases increases. This suggests that adding more data does not necessarily improve extraction accuracy, potentially due to overfitting to noisy examples.

RE Task Optimization. For RE, we conducted the same case number tests and observed that performance is highest at 15 cases but, unlike NER, the F1-score gradually improves again as more cases are added. This indicates that relation extraction benefits from larger training sets. However, considering computational efficiency, we selected 10 cases as the optimal balance between performance and cost. The results are shown in Figure 2.

Demonstration of a Fine-tuning Case. Below is an example of how we format the fine-tuning dataset for NER, using a structured JSONL format. This format follows OpenAI's instruction-tuning style, where the system role defines the task, the user provides input text, and the assistant outputs structured entity annotations.

Listing-10: CONLL04 NER Finetuning dataset

```
{"messages": [
    {"role": "system", "content": "You
        are a helpful assistant that
        extracts Location (LOC), Person
        (PER), and Organization (ORG)
        entities from a given text."},
    {"role": "user", "content": "An art
        exhibit at the Hakawati Theatre
        in Arab east Jerusalem was a
        series of portraits of
        Palestinians killed in the
        rebellion."},
    {"role": "assistant", "content": "{
        \"Person\": [\"NULL\"], \"
        Location\": [\"Jerusalem\"], \"
        Organization\": [\"Hakawati
        Theatre\", \"Arab east Jerusalem
        \"]}"}
]}
```

Listing-11: CONLL04 RE Fine-tuning Dataset

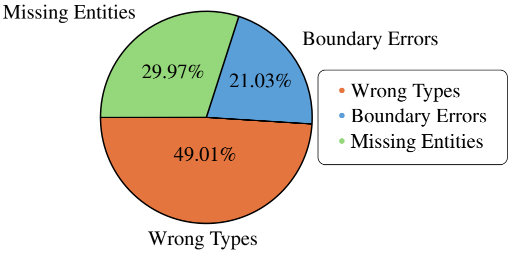

Figure 4: Percentage (%) of different error types in CoNLL-04 for the NER task.

```
{"messages": [
    {"role": "system", "content": "You
        are a helpful assistant tasked
        with extracting relational
        triples (Head, Relation, Tail)
        for the types of relations: Kill
        , Located-in, Live-in,
        Organization-based-in, and Work-
        for from a given text."},
    {"role": "user", "content": "An
        enraged Nikita Khrushchev
        instructed Soviet ships to
        ignore President Kennedy's naval
         blockade during the Cuban
        missile crisis, but the order
        was reversed just hours before
        an inevitable confrontation,
        according to a new book."},
    {"role": "assistant", "content": "{
        \"Relations\": [{\"Head\": \"
        Nikita Khrushchev\", \"Relation
        \": \"Live-in\", \"Tail\": \"
        Soviet\"}]}"}]
}
```

Method	Precision	Recall	F1
Baseline-NER	74.91	65.12	69.68
(+) Summarizer	71.04	73.51	72.25
2nd-round Type-agent	73.02	57.05	64.05

Table 14: Effectiveness (%) of Summarizer Agent (GPT-3.5) on CONLL03

Method	P	R	F1
NER			
-Type-Agent	61.15	78.15	68.61
-CrossAgentIE	64.81	81.34	72.14
-Template finetuning (One LLM)	62.12	82	70.69
RE			
-Type-Agents	57.37	43.98	49.79
-CrossAgentIE-RE	66.10	47.42	55.22
-Template finetuning (One LLM)	37.03	45.12	40.67

Table 15: Performance(%) on CONLL04 with GPT-4o. Precision (P), Recall (R), and F1-score (F1) are reported.

K Error Analysis.

Detailed Error Analysis. As illustrated in Figure 4, the majority of errors in the Baseline-NER stage are **Wrong Types** and **Missing Entities**, together accounting for nearly 80% of all errors. These two categories represent the primary challenges our Type-Agent Multi-Agent Framework seeks to address. The **Wrong Types errors** stem from the GPT-3.5 model's limited ability to distinguish nuanced entity type distinctions within the label set. Even when the entity is correctly identified, the model frequently misclassifies its type due to an inadequate understanding of contextual constraints. Conversely, **Missing Entities** errors often arise from the model's reliance on its pre-trained knowledge base, leading it to prioritize entities that align with prior knowledge while overlooking less frequent or domain-specific entities. This highlights a key limitation in handling entities that deviate from commonly encountered patterns or fall outside the model's pre-trained distribution. To better understand these errors, we further categories Boundary Errors into three subtypes: 1). Contain Gold, where the predicted span fully encompasses the gold entity. 2). Contained by Gold, where the predicted span is entirely within the gold annotation. 3). Overlap with Gold, where the predicted and gold spans partially overlap. By addressing these error types, our framework aims to improve both entity classification and the identification of less-aligned entities, tackling the core sources of failure in the Baseline-NER stage.

Impact of Different Frameworks on Error Types. As shown in Table 8, the proposed 1st-Debate-NER and 2nd-Feedback-NER frameworks introduce distinct improvements across different error types. The Boundary Errors remain relatively stable across all frameworks (Baseline: 90, 1st-Debate: 81, 2nd-Feedback: 90), suggesting that while cross-type debate improves type classification, it does not significantly impact span alignment. Wrong Type Errors, however, show a marked decrease in the 1st-Debate-NER stage (333 → 251), indicating that cross-type debate helps refine entity type classification. Interestingly, these errors increase again in the 2nd-Feedback-NER stage (251 → 343), suggesting that the integration of relation extraction (RE) feedback introduces new type inconsistencies. The most significant improvement is observed in Missing Entities, where the 2nd-Feedback-NER stage reduces errors from 686 (Baseline) to 618, demonstrating that RE feedback enhances recall by recovering previously missed entities. These findings indicate that while cross-type debate enhances type consistency, the RE-NER integration plays a crucial role in entity recovery, shifting the refinement towards higher recall.

Qualitative Error Analysis. Wrong type errors often arise from contextual ambiguity. For example, in "Washington is the capital of the United States," the baseline model misclassified "Washington" as a Person (PER) instead of a Location (LOC) due to statistical biases in pre-trained data. The 1st-Debate-NER framework resolved this by leveraging cross-type discussions, demonstrating its effectiveness in refining entity classification. Boundary errors occur when the predicted span misaligns with the gold annotation. In "The New York Times is a famous newspaper," the baseline model truncated the entity, predicting only "Times" as Organization (ORG) instead of "New York Times." The 1st-Debate-NER framework corrected this by incorporating broader contextual validation, improving span selection. Missing entities remain a challenge in zero-shot settings. In "Barack Obama was elected as the president of the United States," the baseline model failed to detect "Barack Obama" due to low entity prominence in the given context. The 2nd-Feedback-NER framework, through relation-based feedback, successfully recovered the entity by reinforcing contextual dependencies. These cases highlight the strengths of different stages in our framework: cross-type debate improves type consistency, multi-agent validation enhances boundary alignment, and relation-based feedback significantly boosts recall.

Details for Error Correction and Error Increase. In the Baseline-NER stage, errors were dominated by 686 false negatives (FN) and 423 false positives (FP), resulting in a total error count of 1,109. While precision and recall were relatively balanced, the high FP count lowered overall precision and impacted model performance.

With 1st-Debate-NER, false positives dropped significantly from 423 to 332, reducing total errors to 1,012. The primary impact of this stage was an increase in precision, as cross-type debate corrected entity type misclassifications, leading to a modest improvement in the F1-score. However, false negatives (missed entities) remained nearly unchanged, with only a slight reduction from 686 to 680, leading to a minimal recall improvement of 0.22%.

In contrast, the 2nd-Feedback-NER stage focused on recall, reducing false negatives from 680

Method	Time (seconds)	Cost per Doc ID (USD)	Total Tokens
Single Agent	11-14	0.000336	551
Short Conversation (2-4 agents)	18-25	0.000841	1377
Long Conversation (Large Debate)	50-75	0.001682	2755

Table 16: Time and Cost Efficiency of Different Prompting Methods

to 618—a substantial improvement that resulted in an 8.73% increase in recall. However, this gain came at the expense of increased false positives, which rose from 332 to 433, leading to a slight increase in total errors (1,051). Despite this trade-off, the overall F1-score improved, as the reduction in missed entities outweighed the negative impact of additional false positives.

These results highlight the strategic trade-off between precision and recall in an iterative optimization setting. When false negatives dominate the error distribution, a controlled increase in false positives can effectively enhance recall, ultimately leading to better overall performance.

L Time and cost efficiency

Table 16 presents the time, token consumption, and cost per document ID across different settings. The single-agent approach processes each instance in 11-14 seconds with minimal token usage and cost. In contrast, multi-agent interactions (2-4 agents) handling a small number of type labels collaboratively require 18-25 seconds, with token consumption often exceeding twice that of a single agent. More complex scenarios involving over four agents significantly increase computational cost and latency, with conversations lasting 50-75 seconds and token usage rising fourfold or more.

Notably, template fine-tuning—which optimizes a single LLM before inference—achieves efficiency comparable to the single-agent setting, as inference occurs on a fine-tuned model without additional agent interactions, keeping cost and time nearly the same. These findings underscore the trade-offs between efficiency and reasoning complexity, particularly the non-linear cost escalation in multi-agent decision-making.

To quantify the trade-off between performance and inference cost, we introduce an **Efficiency Score metric**, inspired by prior work on computational efficiency in NLP models (Strubell et al., 2019; Kaplan et al., 2020):

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

where F1-score represents the model's accuracy in Named Entity Recognition (NER) or Relation Extraction (RE), and Cost per Doc ID denotes the computational expense (USD) per document. As shown in Table 17, a higher Efficiency Score indicates better cost-effectiveness. Among the evaluated methods, the Single Agent approach achieves the highest Efficiency Score (158.2) due to its extremely low computational cost, despite having the lowest F1-score. This suggests that while it is the most cost-effective in terms of inference expense, its lower accuracy limits its practical utility. In contrast, Template Fine-tuning balances accuracy, inference time, and cost efficiency, achieving a score of 100.70 by significantly improving F1-score while maintaining a relatively low computational cost. CrossAgentIE, although demonstrating strong performance, has the lowest efficiency (60.4) as its higher computational overhead outweighs its accuracy gains.

Method	**Dataset**	**F1-score (%)**	**Cost per Doc ID (USD)**	**Efficiency Score**
Single Agent	CoNLL04	53.13	0.000336	158.2
CROSSAGENTIE	CoNLL04	66.45	0.001100	60.4
Template Fine-tuning	CoNLL04	**70.38**	0.000699	**100.70**

Table 17: Efficiency Score of Different Methods Based on Cost Per Doc_ID