DiZiNER: Disagreement-guided Instruction Refinement via Simulating Pilot Annotation for Zero-shot Named Entity Recognition

Anonymous ACL submission

Abstract

Large language models (LLMs) have advanced information extraction (IE) by enabling zero-shot and few-shot named entity recognition (NER), yet their generative outputs still show persistent and systematic errors. Despite progress through instruction fine-tuning, zero-shot NER still lags far behind supervised systems. These recurring errors mirror inconsistencies observed in early-stage human annotation processes that resolve disagreements through pilot annotation. Motivated by this analogy, we introduce DiZiNER (**Di**sagreement-guided **I**nstruction Refinement via Pilot Annotation Simulation for **Z**ero-shot **N**amed **E**ntity **R**ecognition), a framework that simulates the pilot annotation process, employing LLMs to act as both annotators and supervisors. Multiple heterogeneous LLMs annotate shared texts, and a supervisor model analyzes inter-model disagreements to refine task instructions. Evaluated on 18 benchmarks, DiZiNER achieves new zero-shot SOTA results on 13 datasets, improving prior bests by 9.4 F1 and reducing the zero-shot to supervised gap by over 12 points. It also consistently outperforms its supervisor, GPT-5 mini, indicating that improvements stem from disagreement-guided instruction refinement rather than model capacity. Pairwise agreement between models shows strongly correlation with NER performance, further supporting this finding.[1]

1 Introduction

Information extraction (IE) converts unstructured text into structured data, with named entity recognition (NER) serving as the entry point that identifies and categorizes entity spans. Recent advances in large language models (LLMs) have greatly expanded the potential of IE (Lu et al., 2022; Bogdanov et al., 2024), enabling in-context learning (ICL) strategies for NER such as few-shot (Chen et al., 2023; Jiang et al., 2024) and zero-shot learning (Xie et al., 2023a; Sainz et al., 2023). Despite this progress, state-of-the-art (SOTA) models still depend heavily on human-labeled data, with a wide gap remaining between supervised fine-tuning (SFT) and ICL (Xie et al., 2023a; Naguib et al., 2024).

LLMs exhibit recurring NER error patterns, including difficulty following complex guidelines (Pang et al., 2023; Sainz et al., 2023; Qi et al., 2024), ambiguity in span boundary detection (Guo et al., 2024a; Ding et al., 2024), and frequent confusion of entity types (Li et al., 2024a; Kim et al., 2024). Prior efforts have addressed these issues through instruction fine-tuning on diverse datasets (Wang et al., 2023a), open NER frameworks (Sainz et al., 2023), and large-scale synthetic data generation (Zhou et al., 2023). Yet, supervised methods still outperform them by a considerable margin (Table 2).

In this context, we note that these LLM errors parallel those observed during the early stages of human annotation (Tanabe et al., 2005; Bernier-Colborne and Vajjala, 2024). Gold-standard datasets are typically built through *pilot annotation*, an iterative process of resolving annotator disagreements and refining guidelines (Walker et al., 2006; Weischedel et al., 2011; Finlayson and Erjavec, 2017). Supervisors analyze disagreements, update ambiguous instructions, and align the annotations with downstream application needs (Fort et al., 2009, Figure 1).

Building on this analogy, we propose DiZiNER (**Di**sagreement-guided **I**nstruction Refinement via Pilot Annotation Simulation for **Z**ero-shot **N**amed **E**ntity **R**ecognition), a framework that simulates pilot annotation using LLMs as both annotators and supervisors. Multiple heterogeneous open-source LLMs act as annotators labeling shared texts, and a supervisor LLM analyzes and categorizes inter-

[1]The code and prompts are available at https://anonymous.4open.science/r/diziner-ner-2D58.

model disagreements to refine both shared and model-specific instructions. This iterative cycle of annotation, disagreement analysis, and instruction refinement parallels the workflow of human pilot annotation, allowing LLMs to adapt to individual NER tasks without any parameter updates.

Across 18 NER benchmarks, DiZiNER achieves zero-shot SOTA results on 13 datasets, improving prior bests by +9.4 F1 on average and narrowing the zero-shot-supervised gap from -32.0 to -19.4 points. Agreement metrics between LLM annotators consistently increase across iterations and show a strong correlation with NER performance. Notably, DiZiNER surpasses GPT-4o mini supervisor, indicating that the observed improvements arise from disagreement-guided refinement rather than from the supervisor's inherent capability.

2 Related Works

Instruction tuning for NER Standard instruction fine-tuning often struggles to follow complex annotation guidelines and to produce structured outputs in IE tasks (Qi et al., 2024). InstructUIE and GoLLIE address these challenges by curating NER datasets for instruction fine-tuning, thereby improving zero-shot performance and guideline adherence (Wang et al., 2023b; Sainz et al., 2023). Open NER frameworks relax label constraints, allowing LLMs to better exploit their language understanding capabilities for NER (Etzioni et al., 2011). UniversalNER distills ChatGPT on synthetic data (Zhou et al., 2023), while GLiNER and NuNER adopt encoder-only architectures to reduce inference costs (Zaratiana et al., 2023; Bogdanov et al., 2024). Recent work has sought to unify heterogeneous corpora and to address span ambiguity through boundary-aware learning (Yang et al., 2024; Ding et al., 2024; Guo et al., 2024a). Despite these advances, the performance gap with supervised systems remains large, and reliance on fine-tuning limits adaptation to rapidly evolving LLMs.

Generative NER without instruction tuning In parallel, researchers have explored leveraging LLMs' inherent instruction-following capabilities to perform generative NER without requiring additional instruction fine-tuning. Early work constrained outputs via code-like schema representations (Li et al., 2023; Sainz et al., 2023; Guo et al., 2024b; Li et al., 2024b) or reformulated tagging as token generation (Wang et al., 2023a). Subsequent approaches introduced reasoning-based prompting such as self-consistency and self-verification methods to better convey complex annotation instructions (Xie et al., 2023a; Kim et al., 2024; Pang et al., 2023).

Building on the success of self-consistency and ICL, recent methods for generative NER adopt iterative self-improving strategies by generating and filtering pseudo-examples and providing them as in-context demonstrations (Xie et al., 2023b; Tong et al., 2025). Our work follows this iterative, fine-tuning-free line of research yet distinctly utilizes inter-model disagreement as a signal for improving NER performance, paralleling how human annotators refine guidelines and reconcile judgments during gold-standard dataset construction.

3 DiZiNER

The DiZiNER framework operates through iterative pilot annotation cycles consisting of three stages: **(1) Independent Cross-Annotation**, where multiple LLM annotators independently perform NER tagging on the same set of documents; **(2) Disagreement Analysis**, which identifies *hotspot* spans with high annotation disagreement, categorizes and summarizes disagreement patterns into structured reports; and **(3) Instruction Refinement**, where a supervisor model leverages the resulting structured reports to refine task instructions and reduces inter-model disagreement across iterations.

3.1 Task Formulation

LLM annotators form a heterogeneous pool $\mathcal{M} = \{M_k\}_{k=1}^{K}$ composed of independently developed models to minimize correlated errors. The label set is $\mathcal{L} = \{\ell_i\}_{i=1}^{n}$, and the NER schema is

$$\Sigma = \big\{(\ell,\ d_\ell,\ \mathcal{P}_\ell,\ \mathcal{N}_\ell)\big\}_{\ell \in \mathcal{L}},$$

where d_ℓ is a definition for entity type ℓ, and $\mathcal{P}_\ell, \mathcal{N}_\ell$ are positive and negative examples. The schema Σ remains fixed across iterations to maintain task consistency and prevent task drift.

At iteration t, annotator M_k receives a task configuration

$$\Theta_k^{(t)} = \big(\Sigma,\ C^{(t)},\ R_k^{(t)},\ G^{(t)}\big),$$

where $C^{(t)}$ are shared instructions, $R_k^{(t)}$ are model-specific instructions, and $G^{(t)}$ is the final task goal. Given an input sentence x, the annotator predicts

$$y \sim P_{M_k}\big(y \,|\, x, \Theta_k^{(t)}\big),$$

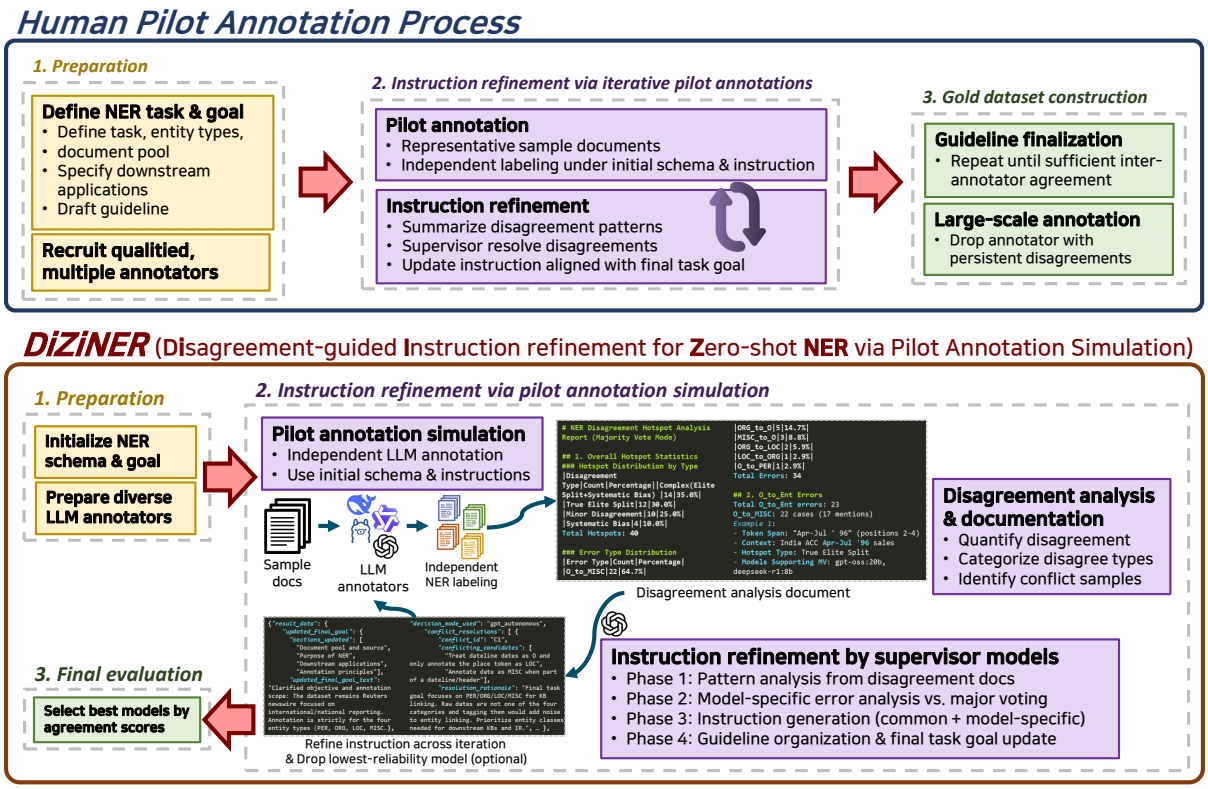

Figure 1: Overview of the DiZiNER framework. Multiple heterogeneous LLMs act as independent annotators. Disagreement profiles are constructed from their outputs, and a supervisor LLM iteratively refines the schema and annotator-specific instructions until convergence.

with labeled outputs $y = \{(e_j,\ \ell_{e_j})\}$, where e_j is an entity span and $\ell_{e_j} \in \mathcal{L}$ its label.

3.2 Independent Cross-Annotation

At each iteration, documents are grouped by lexical diversity, and a representative subset is randomly sampled across groups to form the iteration document set $\mathcal{D}^{(t)}$. All annotators in $\mathcal{M}$ independently label each sample in the set according to their task configuration $\Theta_k^{(t)}$. To enable token-level comparison across models, span-level annotations are converted into a BIO sequence representation. For input $x = (w_1, \ldots, w_m)$, the tag set is defined as

$$\mathcal{T} = \{\mathrm{B}{-}\ell,\ \mathrm{I}{-}\ell,\ \mathrm{O} \mid \ell \in \mathcal{L}\}.$$

The conversion yields a BIO sequence

$$\mathbf{z}_k(x) = (z_{k,1}(x), \ldots, z_{k,m}(x)), \quad z_{k,i}(x) \in \mathcal{T},$$

representing the token-level tagging output derived from the span-level annotation y of annotator M_k.

3.3 Disagreement Analysis

This stage identifies *hotspot* spans that exhibit strong inter-model disagreement. Token-level inconsistencies across annotators are quantified to mark high-disagreement regions.

Model Weights and Consensus Model weights are computed from pairwise strict span F1 scores between annotators, where for models M_i and M_j,

$$\mathrm{F1}_{ij} = \frac{2\,|\mathcal{S}_i \cap \mathcal{S}_j|}{|\mathcal{S}_i| + |\mathcal{S}_j|},$$

where $\mathcal{S}_k$ denotes the set of predicted entity spans from model M_k. Each model's weight, w_k, is computed as the average of its pairwise F1 scores with all others, normalized so that the weights sum to one. The *elite set* is defined as the subset of annotators with the highest weights whose cumulative sum first attains 0.5 when sorted in descending order. The computed model weights are also used as each annotator's agreement score in subsequent analyses.

The consensus label for token i in sentence x is obtained via weighted majority voting,

$$\widehat{\tau}(x, i) = \arg\max_{\tau \in \mathcal{T}} p_\tau(x, i),$$

where $p_\tau(x, i) = \sum_k w_k\, \mathbf{1}[z_{k,i}(x) = \tau]$ represents the weighted token-wise probability for tag τ.

Hotspot Span Identification We compute three complementary token-level measures capturing distinct forms of annotation disagreement. (1) *Label conflict* quantifies dispersion among BIO tags,

$$D_{\text{conf}}(x,i) = 1 - \sum_{\tau \in \mathcal{T}} p_\tau(x,i)^2.$$

(2) *Type confusion* reflects disagreement over entity types,

$$D_{\text{type}}(x,i) = 1 - \sum_{\ell \in \mathcal{L} \cup \{\text{O}\}} \big(p_{\text{B}-\ell}(x,i) + p_{\text{I}-\ell}(x,i)\big)^2.$$

(3) *Boundary uncertainty* measures inconsistency at entity boundaries,

$$U_{\text{bnd}}(x,i) = \max\Big\{4p_{\text{B}}(x,i)\big(1 - p_{\text{B}}(x,i)\big), \\ 4p_{\text{I}}(x,i)\big(1 - p_{\text{I}}(x,i)\big)\Big\}.$$

The final token-level disagreement score is defined as

$$U_\star(x,i) = \max\{D_{\text{conf}},\, D_{\text{type}},\, U_{\text{bnd}}\}.$$

Tokens are ranked by their $U_\star(x,i)$ scores, and the top 20% are identified as high-disagreement regions. Neighboring tokens in this range are merged into hotspot spans, which are subsequently flagged for supervisor review and used for refining instructions.

Documentation Each iteration produces a summary report detailing hotspot statistics and model differences between elite and non-elite groups, outlining disagreement types and error categories (*O→Ent*, *Ent→O*, *Ent→Ent*, *Span Error*). Representative examples with brief reasoning traces demonstrate characteristic disagreement patterns that inform targeted instruction refinement.

3.4 Instruction Refinement

The supervisor model iteratively refines task instructions based on disagreement documents and instructions from the previous iteration to improve annotator agreement. Each cycle proceeds through four phases:

1. **Disagreement pattern analysis.** Identify recurring disagreement patterns in hotspot summaries and infer their underlying causes. Extract generalizable correction principles rather than case-specific fixes (Table 8).
2. **Model-specific diagnosis.** Examine residual errors for each non-elite model using the consensus output $\widehat{\tau}(x,i)$, excluding patterns already addressed in Phase 1. Identify model-specific weaknesses and formulate targeted adjustments (Table 9).
3. **Guideline integration and conflict resolution.** Integrate the refined instructions from the current supervision cycle with those from previous iterations, resolving any conflicts based on the final task goal, which aims to maximize performance in downstream applications (Table 10).
4. **Hierarchical organization.** Reorganize refined instructions into a hierarchical structure where general rules precede specific or conditional cases. This restructuring enhances clarity and readability (Table 11).

A small set of tuning parameters was introduced to regulate the stability of iterative updates, and three parameter configurations were explored to ensure consistency across heterogeneous benchmarks (see Appendix A). For supervised ablations, the same procedure is executed with gold-standard labels replacing the consensus outputs.

3.5 Identification of Best Model Configuration

DiZiNER selects the optimal iteration–model configuration in the absence of human-labeled data, ensuring an entirely zero-shot setting. We observe that pairwise annotator agreement, measured by strict span-level F1, is strongly correlated with NER performance (Figure 2), and thus use agreement statistics to guide configuration selection. Accordingly, all iteration–model pairs are ranked by their mean agreement, and the top three candidates are evaluated on the test set.

4 Experiments

4.1 Settings

Datasets We evaluate our framework on a total of 18 NER datasets spanning diverse domains, including the CrossNER suite (AI, Literature, Music, Politics, and Science; Liu et al., 2021), general-purpose corpora such as CoNLL2003 (Sang and De Meulder, 2003), ACE2005 (Walker et al., 2006), OntoNotes (Pradhan et al., 2013), and MultiNERD (Tedeschi and Navigli, 2022); biomedical corpora including AnatEM (Pyysalo and Ananiadou, 2014),

Methods	AI	Literature	Music	Politics	Science	Average
ChatGPT (Zhou et al., 2023)	52.4	39.8	66.6	68.5	67.0	58.9
GPT-4 (Yang et al., 2024)	50.0	55.2	59.2	63.4	63.2	58.2
InstructUIE (Wang et al., 2023b)	49.0	47.2	53.2	48.1	49.2	49.3
UniNER-7B (Zhou et al., 2023)	53.6	59.3	67.0	60.9	61.1	60.4
UniNER-13B (Zhou et al., 2023)	54.2	60.9	64.5	61.4	63.5	60.9
GLiNER (Zaratiana et al., 2023)	57.2	64.4	69.6	72.6	62.6	65.3
GoLLIE (Sainz et al., 2023)	61.6	62.7	68.4	60.2	56.3	61.8
KnowCoder-7B (Li et al., 2024b)	60.3	61.1	70.0	72.2	59.1	64.5
IRRA (Xie et al., 2024)	57.5	59.3	69.4	74.0	68.3	65.7
GNER (Ding et al., 2024)	68.2	68.7	81.2	75.1	76.7	74.0
B2NER (Yang et al., 2024)	64.7	71.6	82.4	78.2	**79.4**	75.3
GPT-5 mini (supervisor model)	64.3	67.6	73.3	72.8	68.4	69.3
DiZiNER	**71.1**	**73.8**	**82.9**	**80.9**	75.4	**76.8**
Avg. Gain from Iteration 0	**+2.7**	**+3.6**	**+11.1**	**+2.2**	**+4.5**	**+4.8**
$\Delta_{\textit{DiZiNER - GPT-5 mini}}$	**+6.8**	**+6.2**	**+9.6**	**+8.1**	**+7.0**	**+7.5**
$\Delta_{\textit{DiZiNER - Best Prior Zero-shot}}$	**+2.9**	**+2.2**	**+0.5**	**+2.7**	**-4.0**	**+0.9**

Table 1: Zero-shot NER performance on the CrossNER dataset. Best per domain in **bold**, second-best underlined. Avg. Gain from Iteration 0 denotes the average improvement averaged across eight annotator models, computed as the mean difference between each model's Iteration-0 score and its best-performing iteration within the iterative document set.

BC2GM (Smith et al., 2008), BC4CHEMD (Wang et al., 2019), BC5CDR (Li et al., 2016), and GENIA (Kim et al., 2003); a STEM-oriented corpus FabNER (Kumar and Starly, 2022); and social or conversational datasets such as BroadTwitter (Derczynski et al., 2016), MIT-Movie, and MIT-Restaurant (Liu et al., 2013). To simulate pilot annotation, we use the training splits to refine task instructions, while the final performance was evaluated on the corresponding test sets (Table 7).

Baselines We compare DiZiNER against representative baselines under both *zero-shot* and *supervised* settings. The zero-shot setting excludes models that rely on task-specific fine-tuning or retrieval-based ICL, namely **ChatGPT** (Zhou et al., 2023), **GPT-4** (Yang et al., 2024), **InstructUIE** (Wang et al., 2023b), **UniNER-7B/13B** (Zhou et al., 2023), **GLiNER** (Zaratiana et al., 2023), **GoLLIE** (Sainz et al., 2023), **KnowCoder-7B** (Li et al., 2024b), **GNER** (Ding et al., 2024), **B2NER** (Yang et al., 2024), **IRRA** (Xie et al., 2024), **EvoPrompt** (Tong et al., 2025), and **GPT-5 mini**. For the supervised setting, we include SFT models trained on gold annotations, including **BERT-base** and **InstructUIE** (Wang et al., 2023b), **UniNER**, **GLiNER**, **KnowCoder-7B**, **GNER**, and **B2NER**.

Backbones and Implementation DiZiNER employs a heterogeneous pool of eight open-source LLMs, each independently developed by different organizations with distinct training architectures, datasets, and optimization pipelines and accessed via OpenRouter[2]: `mistral-small3.2:24b`, `gpt-oss:20b`, `phi4:14b`, `qwen3:14b`, `gemma3:12b`, `deepseek-r1:8b`, `llama3.1:8b`, `nemotron-nano:8b`. This diversity promotes independent judgment among annotators and minimizes correlated errors. The supervisor model was `GPT-5-mini-2025-08-07`, accessed via the OpenAI API between August 7 and September 30, 2025.

Each iteration processes a document set of 25 samples, with up to five refinement cycles. Three parameter configurations are explored to ensure consistent application across heterogeneous benchmarks (Table 6).

Metrics We report the entity-level micro-F1 under the strict span setting as our evaluation metric, requiring both entity boundary and type to be correctly predicted.

[2] https://openrouter.ai

Methods	Movie	Rest.	B-Twit	ACE05	CoNLL	M-NERD	Onto	FabNER	Anat	bc2	bc4	bc5	GENIA	Avg
Zero-shot														
ChatGPT (Zhou et al., 2023)	5.3	32.8	61.8	26.6	52.5	58.1	29.7	15.3	30.7	40.2	35.5	52.4	41.6	37.1
GoLLIE (Sainz et al., 2023)	63.0	52.7	51.4	–	–	77.5	–	26.3	–	–	–	–	–	–
UniNER-7B (Zhou et al., 2023)	42.4	31.7	67.9	36.9	72.2	59.3	27.8	24.8	25.1	46.2	47.9	68.0	54.1	46.5
GLiNER (Zaratiana et al., 2023)	57.2	42.9	61.2	27.3	64.6	59.7	32.2	23.6	33.3	47.9	43.1	66.4	55.5	47.3
EvoPrompt (Tong et al., 2025)	70.9	**69.3**	–	**51.2**	81.3	–	–	–	–	–	–	–	–	–
GPT-5 mini (supervisor model)	73.3	58.5	59.2	54.0	81.8	74.0	63.8	29.5	59.2	73.0	63.7	62.8	56.5	62.3
DiZiNER (Zero-shot)	**78.5**	68.1	**78.3**	46.2	**88.6**	**83.7**	**63.5**	**29.1**	**60.7**	**73.0**	**81.7**	**81.3**	**60.9**	**68.7**
Avg. Gain from Iteration 0	**+5.6**	**+22.9**	**+20.1**	**+2.1**	**+28.3**	**+2.6**	**+24.8**	**+0.9**	**+25.3**	**+12.7**	**+26.9**	**+16.4**	**+4.5**	**+14.9**
$\Delta_{DiZiNER\ -\ GPT\text{-}5\ mini}$	**+5.2**	**+8.6**	**+19.1**	**-7.8**	**+6.8**	**+9.7**	**-0.3**	**-0.4**	**+1.5**	**+0.0**	**+18.0**	**+18.5**	**+4.4**	**+6.4**
Supervised														
BERT-base (Wang et al., 2023b)	88.8	81.0	58.6	**87.3**	92.4	91.3	91.1	64.2	85.8	80.9	86.7	85.3	73.3	82.1
InstructUIE (Wang et al., 2023b)	89.6	82.6	80.3	79.9	91.5	90.3	88.6	78.4	88.5	80.7	87.6	89.0	75.7	84.8
UniNER-7B (Zhou et al., 2023)	90.2	82.3	81.2	86.7	93.3	93.7	89.9	81.9	88.5	82.4	89.2	89.3	77.5	**86.6**
GLiNER (Zaratiana et al., 2023)	87.9	83.6	**82.7**	82.8	92.6	**93.8**	89.0	77.8	88.9	83.7	87.9	88.7	**78.9**	86.0
KnowCoder-7B (Li et al., 2024b)	90.6	81.3	78.3	86.1	**95.1**	93.1	88.2	82.9	86.4	82.0	–	89.3	76.7	–
GNER (Ding et al., 2024)	90.2	**83.8**	81.3	–	93.6	94.4	**91.8**	**85.4**	**90.3**	**84.3**	**90.0**	**90.3**	–	–
B2NER (Yang et al., 2024)	**90.8**	83.7	82.2	83.0	92.6	94.0	84.3	78.8	89.2	82.0	89.0	88.5	76.4	85.7
$\Delta_{Best\ prior\ ZS\ -\ Best\ prior\ Sup.}$	**-19.9**	**-14.5**	**-14.8**	**-36.1**	**-13.8**	**-16.9**	**-59.6**	**-59.1**	**-57.0**	**-36.4**	**-42.1**	**-22.3**	**-23.4**	**-32.0**
$\Delta_{DiZiNER\ -\ Best\ prior\ ZS}$	**+7.6**	**-2.2**	**+10.4**	**-5.0**	**+7.3**	**+6.2**	**+31.3**	**+2.8**	**+27.4**	**+25.1**	**+33.8**	**+13.3**	**+5.4**	**+12.6**
$\Delta_{DiZiNER\ -\ Best\ prior\ Sup.}$	**-12.3**	**-16.7**	**-4.4**	**-41.1**	**-6.5**	**-10.7**	**-28.3**	**-56.3**	**-29.6**	**-11.3**	**-8.3**	**-9.0**	**-18.0**	**-19.4**

Table 2: Overall NER results across 14 benchmarks. *ZS* denotes our zero-shot pipeline without any gold labels, and all DiZiNER results are zero-shot. Within each setting (zero-shot and supervised), the best and second-best scores are highlighted in **bold** and underlined, respectively. GPT-5 results are excluded from this comparison. Avg. Gain from Iteration 0 denotes the average improvement averaged across eight backbone models, computed as the mean difference between each model's Iteration-0 score and its best-performing iteration within the iterative document set. Abbreviations: Movie = MIT-Movie, Rest. = MIT-Restaurant, B-Twit = BroadTwitter, ACE05 = ACE2005, CoNLL = CoNLL2003, M-NERD = MultiNERD, Onto = OntoNotes, Anat = AnatEM, bc2 = BC2GM, bc4 = BC4CHEMD, bc5 = BC5CDR.

4.2 Main Results

Without any instruction fine-tuning, DiZiNER establishes new zero-shot SOTA results on 15 out of 18 benchmarks (Tables 1 and 2). On CrossNER, DiZiNER achieves SOTA performance in four of the five domains, excluding *Science*, with an average F1 of 76.8, outperforming B2NER (Yang et al., 2024) by +1.5 and the previous best zero-shot system by +0.9 F1 across domains. In addition, compared with its GPT-5 mini supervisor, DiZiNER yields an average improvement of +7.5 F1 (Table 1).

Across benchmarks with available supervised results, DiZiNER improves the average zero-shot performance by +12.6 F1 over the best prior zero-shot systems and narrows the gap between zero-shot and supervised performance from -32.0 to -19.4 F1 (Table 2). DiZiNER surpasses its GPT-5 mini supervisor by an average of +6.4 F1, demonstrating that the observed improvements arise from disagreement-guided refinement rather than the supervisor's intrinsic capability, even though all annotator models are open-source and under 24B parameters.

In practice, NER performance within the training samples consistently improved across iterations. When averaged over the eight annotator models, performance increased from Iteration 0 to each model's best-performing iteration by as much as +20 F1 on several benchmarks, with an overall average gain of +14.9 F1 (Table 2) and +4.8 F1 on CrossNER (Table 1). On average, the best-performing configurations appeared around 2.7 iterations.

Moreover, we observed a strong positive correlation between the gold-standard F1 measured within each iteration's document set and the model's average pairwise agreement across refinement cycles (Figure 2). This relationship shows that higher agreement consistently corresponds to higher gold-standard performance, indicating that disagreement among LLM annotators is a reliable signal of underlying task quality and providing empirical support for the effectiveness of the proposed DiZiNER framework.

The average cost per iteration was $1.90 for inference and $0.77 for supervision, resulting in a total of $2.67 per iteration. Considering that an average of five iterations were conducted for each benchmark and three configuration settings were explored, the total cost amounts to $40.1 per benchmark.

4.3 Ablation Study

Final task goal Skipping the final task goal consistently reduced performance (Table 3). In our framework, this component was designed to serve as a global criterion that guides instruction refinement toward the overall task objective. We speculate that when it is omitted, the refined instructions may remain locally consistent yet diverge from the benchmark's intended direction, leading to lower F1 scores across domains.

Benchmark	Base	+sfg
AI	**76.4**	75.5
Literature	**76.7**	75.5
Music	**89.1**	85.2
Politics	**85.9**	85.6
Science	**74.8**	**74.8**
CoNLL2003	**87.1**	74.6
Average	**83.1**	79.4

Table 3: Best NER performance with and without skipping the final task goal (+sfg). Bold indicates higher performance per benchmark.

Annotator diversity To assess the importance of model diversity, we constructed single-family annotator pools using only Llama or Qwen models, each containing eight models ranging from 8B to 70B parameters (Table 4). While the inclusion of larger models sometimes led to comparable or slightly higher absolute F1 scores, their best results consistently appeared at iteration 0, suggesting that disagreement-guided refinement failed to take effect. This finding underscores the importance of inter-model diversity for enabling effective refinement and sustained performance gains.

Iteration set size We examined how the number of documents per iteration influences refinement. Optimal performance was achieved with 15-20 samples (Table 5), while larger sets degraded results by expanding hotspot regions and obscuring distinct error patterns. Further scaling of the iteration document set size appears unnecessary under the current framework.

Removing the least consistent annotator The effect of excluding the most disagreement-prone annotator varied across benchmarks with no consistent trend (Table 12). Removing the least consistent annotator sometimes improved results but also risked destabilizing disagreement statistics by reducing diversity. Given these trade-offs, we treat this step as a tunable option rather than a fixed rule.

DiZiNER with gold-standard data Incorporating gold supervision provided minimal benefits in our framework (Table 13). Average performance decreased by 0.4 F1, with consistent gains observed only on ACE05 (+9.6), where human annotations helped resolve errors arising from missing context and pronominal references. Replacing disagreement signals with gold labels shifted the objective from cross-model consensus to fixed-target fitting, thereby reducing diversity and weakening iterative refinement. Overall, the disagreement-guided setup without supervision achieved greater stability and stronger performance.

5 Conclusion

We introduce DiZiNER, a zero-shot NER framework that simulates human pilot annotation through disagreement-guided instruction refinement without any parameter updates. By employing multiple heterogeneous LLMs as annotators and a supervisor model for disagreement-driven refinement, DiZiNER reduces boundary ambiguity and type confusion. Across 18 benchmarks, it achieves zero-shot SOTA results on 15 datasets, improves over the previous best zero-shot systems by +12.6 F1 on average, outperforms the GPT-5 mini supervisor, and narrows the zero-shot to supervised gap from -32.0 to -19.4 F1. Ablation studies show that aligning refinement with the final task objective is essential for resolving conflicting instructions and that annotator diversity is critical for effective updates. The strong correlation between agreement metrics and gold-standard F1 indicates that disagreement-guided refinement is the primary driver of gains, suggesting that small open-source models can surpass advanced proprietary baselines in an entirely zero-shot setting without instruction fine-tuning.

Limitations

Our framework exhibits varying gains across benchmarks. This variability likely stems from stochasticity and sampling differences that can alter the trajectory of iterative refinement. Because DiZiNER represents each dataset through its document pool and NER schema without accessing gold-labeled samples, refined instructions may gradually drift from dataset-specific annotation conventions. A hybrid approach that combines disagreement-guided

Benchmark	Base		Qwen		Llama	
	F1 (%)	Iter / Model	F1 (%)	Iter / Model	F1 (%)	Iter / Model
CrossNER-AI	76.4	2 / mistral-small3.2:24b	**77.5**	0 / qwen2.5-coder:32b	76.7	0 / llama3.3:70b
CrossNER-Literature	**76.7**	2 / qwen3:14b	75.2	0 / qwen2.5-coder:32b	73.7	0 / llama3.3:70b

Table 4: Best zero-shot NER performance with different annotator families. The Base configuration combines eight heterogeneous model families, whereas Qwen and Llama use eight models from a single family each. Iteration number and best-performing model are reported. Bold denotes the highest value per benchmark.

Benchmark	Iteration Document Set Size			
	10	15	25	50
CrossNER-AI	76.2	**77.5**	69.2	70.1
CrossNER-Literature	**85.1**	76.7	73.0	71.3
Average	**80.7**	77.1	71.1	70.7

Table 5: Zero-shot NER performance across different iteration document set sizes.

refinement with a small number of supervised examples could anchor the process to the intended labeling criteria while preserving the efficiency and generality of zero-shot learning.

We also keep the NER schema fixed across iterations to maintain comparability and evaluation consistency with benchmark datasets. This design departs from realistic pilot annotation workflows, where entity types are often added, merged, or removed to resolve ambiguities and better capture domain semantics. Extending DiZiNER to real-world corpus construction will therefore require a schema-refinement component that can propose, test, and validate type updates while ensuring backward compatibility with earlier iterations and preserving evaluation continuity.

References

Gabriel Bernier-Colborne and Sowmya Vajjala. 2024. Annotation errors and ner: A study with ontonotes 5.0. *arXiv preprint arXiv:2406.19172*.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data. *arXiv preprint arXiv:2402.15343*.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023. Learning in-context learning for named entity recognition. *arXiv preprint arXiv:2305.11038*.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 1169–1179.

Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Bowen Yan, and Min Zhang. 2024. Rethinking negative instances for generative named entity recognition. *arXiv preprint arXiv:2402.16602*.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10.

Mark A Finlayson and Tomaž Erjavec. 2017. Overview of annotation creation: Processes and tools. In *Handbook of linguistic annotation*, pages 167–191. Springer.

Karën Fort, Maud Ehrmann, and Adeline Nazarenko. 2009. Towards a methodology for named entities annotation. In *Linguistic Annotation Workshop*, pages 142–145.

Quanjiang Guo, Yihong Dong, Ling Tian, Zhao Kang, Yu Zhang, and Sijie Wang. 2024a. Baner: Boundary-aware llms for few-shot named entity recognition. *arXiv preprint arXiv:2412.02228*.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and 1 others. 2024b. Retrieval-augmented code generation for universal information extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 30–42. Springer.

Guochao Jiang, Zepeng Ding, Yuchen Shi, and Deqing Yang. 2024. P-icl: Point in-context learning for named entity recognition with large language models. *arXiv preprint arXiv:2405.04960*.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

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

Aman Kumar and Binil Starly. 2022. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33(8):2393–2407.

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

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023. Codeie: Large code generation models are better few-shot information extractors. *arXiv preprint arXiv:2305.05711*.

Yuepei Li, Kang Zhou, Qiao Qiao, Qing Wang, and Qi Li. 2024a. Re-examine distantly supervised ner: a new benchmark and a simple approach. *arXiv preprint arXiv:2402.14948*.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, and 1 others. 2024b. Knowcoder: Coding structured knowledge into llms for universal information extraction. *arXiv preprint arXiv:2403.07969*.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13452–13460.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. *arXiv preprint arXiv:2203.12277*.

Marco Naguib, Xavier Tannier, and Aurélie Névéol. 2024. Few-shot clinical entity recognition in english, french and spanish: masked language models outperform generative model prompting. *arXiv preprint arXiv:2402.12801*.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. *arXiv preprint arXiv:2310.05066*.

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.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2024. Adelie: Aligning large language models on information extraction. *arXiv preprint arXiv:2405.05008*.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv preprint arXiv:2310.03668*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Larry Smith, Lorraine K Tanabe, Rie Johnson Nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, and 1 others. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(Suppl 2):S2.

Lorraine Tanabe, Natalie Xie, Lynne H Thom, Wayne Matten, and W John Wilbur. 2005. Genetag: a tagged corpus for gene/protein named entity recognition. *BMC bioinformatics*, 6(Suppl 1):S3.

Simone Tedeschi and Roberto Navigli. 2022. Multinerd: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 801–812.

Zeliang Tong, Zhuojun Ding, and Wei Wei. 2025. Evoprompt: Evolving prompts for enhanced zero-shot named entity recognition with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5136–5153.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *(No Title)*.

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

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, and 1 others. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2019. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, and 1 others. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 17.

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

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023b. Self-improving for zero-shot named entity recognition with large language models. *arXiv preprint arXiv:2311.08921*.

Tingyu Xie, Jian Zhang, Yan Zhang, Yuanyuan Liang, Qi Li, and Hongwei Wang. 2024. Retrieval augmented instruction tuning for open ner with large language models. *arXiv preprint arXiv:2406.17305*.

Yuming Yang, Wantong Zhao, Caishuang Huang, Junjie Ye, Xiao Wang, Huiyuan Zheng, Yang Nan, Yuran Wang, Xueying Xu, Kaixin Huang, and 1 others. 2024. Beyond boundaries: Learning a universal entity taxonomy across datasets and languages for open named entity recognition. *arXiv preprint arXiv:2406.11192*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv preprint arXiv:2308.03279*.

A Tuning Parameters for Instruction Refinement

To stabilize instruction refinement and prevent excessive corrections during iterative updates, the supervisor employs a series of tuning parameters. Each parameter controls a distinct aspect of the refinement process, ensuring balanced evolution of the instruction set across different phases.

Parameter Definitions

- `max_shared_instructions` Specifies the maximum number of newly generated shared principles per iteration. This prevents uncontrolled guideline expansion and is primarily active in **Phase 1** and **Phase 3**.
- `max_patterns` Determines the number of disagreement patterns considered in each cycle. By focusing only on the most recurrent inconsistencies, it guides efficient refinement during **Phase 1**.
- `max_model_specific_instructions` Sets an upper bound on model-specific adjustments per annotator model. This maintains an appropriate balance between general and specialized rules in **Phase 2** and **Phase 3**.
- `limit_instruction_changes` Enables a controlled edit mode that constrains the degree of revision between refinement cycles, applied during **Phase 4**.
- `max_change_ratio` When controlled editing is active, this parameter limits the proportion of textual modifications to preserve continuity and prevent semantic drift, also enforced in **Phase 4**.

Representative Configurations Three parameter configurations were adopted across benchmarks to investigate varying levels of refinement adaptiveness (Table 6). All settings used a group size of 25 samples per iteration and a maximum of four refinement cycles.

B Dataset Statistics

Table 7 summarizes the datasets used across experiments, encompassing 18 NER benchmarks from general, biomedical, STEM, and social domains to ensure broad domain coverage and diversity of entity types.

C Prompts

Tables 8-11 summarize the four supervisory prompts used for instruction refinement: disagreement analysis, model-specific error review, instruction generation, and hierarchical organization. Each phase builds on the previous to ensure consistent, interpretable NER annotation. Full prompt templates and JSON schema are available on the project's GitHub repository.

D Supplementary Results

Figure 2 visualizes the correlation between inter-annotator agreement and NER performance across refinement iterations. Table 5 summarizes the results under different document set sizes per iteration. Table 12 shows the comparison with and without the removal of the least consistent annotator. Table 13 provides a comparison between zero-shot and supervised evaluation settings.

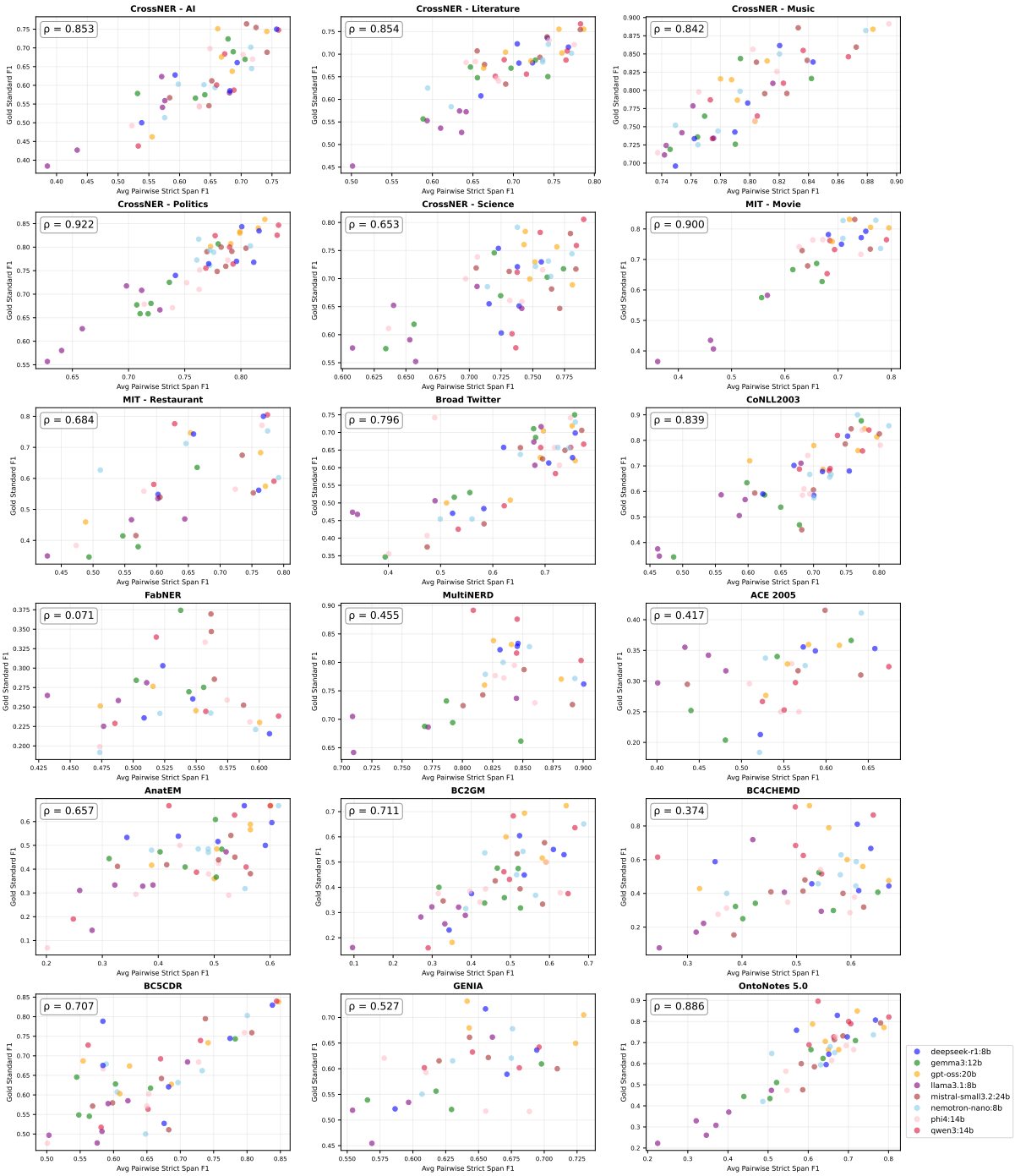

Figure 2: Correlation between pairwise agreement and NER performance across training iterations.Each subplot represents individual benchmarks, showing the Pearson correlation (ρ) between inter-annotator agreement (x-axis) and F1 performance on iteration document sets (y-axis). Strong positive correlations across diverse domains confirm that agreement statistics serve as a reliable, label-free indicator of NER performance during DiZiNER cycles.

Configuration	max_shared	max_patterns	max_model_spec.	limit_changes	max_ratio	max_iter.
Stable	3	5	2	True	0.10	5
Relaxed	5	8	3	False	0.20	5
Aggressive	10	20	10	False	0.50	5

Table 6: Tuning configurations for instruction refinement experiments.

Domain	Dataset	# train	# dev	# test	# types	Avg. tokens	Avg. entities
Cross-domain (CrossNER)	AI (Liu et al., 2021)	100	350	431	14	52	5.3
	Literature (Liu et al., 2021)	100	400	416	12	54	5.4
	Music (Liu et al., 2021)	100	380	465	13	57	6.5
	Politics (Liu et al., 2021)	199	540	650	9	61	6.5
	Science (Liu et al., 2021)	200	450	543	17	54	5.4
Social Media / Dialogue	MIT-Movie (Liu et al., 2013)	9775	2442	2443	12	10	2.2
	MIT-Restaurant (Liu et al., 2013)	7660	1520	1521	8	9	2.0
	BroadTwitter (Derczynski et al., 2016)	5334	2001	2000	3	28	0.5
General	ACE2005 (Walker et al., 2006)	7299	971	1060	7	21	2.8
	CoNLL2003 (Sang and De Meulder, 2003)	14041	3250	3453	4	25	2.8
	MultiNERD (Tedeschi and Navigli, 2022)	134144	10000	10000	16	28	1.6
	OntoNotes (Pradhan et al., 2013)	59924	8528	8262	18	18	0.9
STEM	FabNER (Kumar and Starly, 2022)	9435	2182	2064	12	36	5.1
Biomedical	AnatEM (Pyysalo and Ananiadou, 2014)	5861	2118	3830	1	37	0.7
	BC2GM (Smith et al., 2008)	12500	2500	5000	1	36	0.4
	BC4CHEMD (Wang et al., 2019)	30682	30639	26364	1	45	0.9
	BC5CDR (Li et al., 2016)	4560	4581	4797	2	41	2.2
	GENIA (Kim et al., 2003)	15022	1669	1855	5	46	3.5

Table 7: Statistics of datasets used in our experiments. We evaluate across 18 NER datasets covering general, biomedical, STEM, and social domains.

Prompt Text

You are a strict, methodical NER annotation supervisor. Your task in this phase is to analyze disagreement patterns using majority-vote as a reference point (NOT as ground truth), identify high-yield error patterns, and classify them systematically. Focus on extracting actionable patterns that can inform instruction creation without generating instructions yet.
Current NER Scheme: `{current_ner_schema}`.
Final Task Goal: `{final_task_goal}`.
Disagreement Analysis: `{NER_disagreement_summaries}`.
Task: 1. Analyze disagreement patterns using MV as reference point, acknowledging that MV is not ground truth but a useful consensus measure. 2. Identify and quantify disagreement patterns, clustering them into maximum 8 high-impact categories. 3. For each pattern, determine root causes and assess whether existing instructions already address them. 4. Extract possible annotation approaches for each conflicting case, providing the rationale behind each approach. 5. Identify aspects of the final task goal that need clarification to resolve ambiguous annotation choices. 6. Do NOT generate instructions in this phase - focus on pattern analysis and candidate instruction principles.
Output Format (JSON):

```
{
  "disagreement_analysis_summary": {
    "major_disagreement_sources": [
      "Source 1", ...
    ],
    "mv_reference_reliability": "Assessment of MV as reference point",
    "elite_vs_non_elite_patterns":
    "Comparison between elite and non-elite model behaviors"
  },
  "identified_patterns": [
    {
      "pattern_id": "P1",
      "pattern_name": "Descriptive pattern name",
      "frequency": "high|medium|low",
      "disagreement_subtypes": [
        "Subtype A", ...
      ],
      "root_cause_analysis":
      "Fundamental principle-level explanation of disagreement source",
      "affected_entity_types": [
        "PER", ...
      ],
      "annotation_approaches": [
        {
          "approach": "Annotation approach A",
          "rationale": "Why this approach makes sense",
          "supporting_models": [
            "model1", ...
          ]
        }, ...
      ], ...
  ]
}
```

Table 8: Prompt for instruction refinement in phase 1 (Disagreement Pattern Analysis)

Prompt Text

You are a strict, methodical NER annotation supervisor. In this phase, you analyze error patterns specific to `model_name` based on the configuration, focusing on systematic deviations not covered by common disagreement patterns identified in Phase 1. IMPORTANT: You are analyzing only one model at a time, not multiple models.

Inputs

- Phase 1 results: `phase1_results`
- Single model detailed disagreement data: `model_disagreement_data`
- Elite model identification results: `elite_models`
- Single model bias analysis: `model_bias_analysis`
- Original NER scheme: `current_ner_schema`
- Final task goal: `final_task_goal`
- Existing model-specific instructions: `existing_model_instructions`

Runtime Identifiers: `model_name`, `model_type`

Task Focus

- Identify model-unique patterns not covered by Phase 1
- Classify into `systematic_bias | confusion_pattern | under_tagging | over_tagging | boundary_errors`
- Assess existing model-specific instructions and their effectiveness
- Prepare instruction *needs* (do not generate instructions yet)

Output Format (JSON):

```
{
  "model_name": "{model_name}",
  "elite_or_not": true|false,
  "model_specific_patterns": [
    {
      "pattern_id": "M1_{model_name}",
      "pattern_name": "Model-specific pattern description",
      "pattern_type": "systematic_bias|confusion_pattern|...",
      "not_covered_by_common_patterns": true,
      "pattern_characterization": "How this model behaves differently",
      "examples": [
        "Context - MV: PER(\"...\"), {model_name}: MISC(\"...\")", ...
      ],
      "existing_instruction_assessment": {
        "covered_by_existing_model_specific": "true|false",
        "existing_instruction_reference": "instruction_id or null",
        "effectiveness_assessment": "qualitative note"
      }
    }
  ],
  "model_bias_summary": {
    "primary_systematic_biases": ["Bias 1", "Bias 2"],
    "model_strengths": ["Strength 1", "Strength 2"],
    "key_weaknesses": ["Weakness 1", "Weakness 2"],
    "deviation_from_mv_coalition": "narrative summary"
  }, ...
  "instruction_candidate_needs_max": "{max_model_specific_instructions}"
}
```

Table 9: Prompt for instruction refinement in phase 2 (Single-Model Error Analysis)

Prompt Text

You are a strict, methodical NER annotation supervisor acting as an instruction generator and conflict resolver. Convert identified patterns into concrete instructions, resolve conflicts using the final task goal guidance, and run in either human-interactive or GPT-autonomous mode.

Inputs

- Phase 1 results: `phase1_results`
- Phase 2 results: `phase2_results`
- Final task goal: `final_task_goal`
- Decision mode: `decision_mode` (`human_interactive`|`gpt_autonomous`)
- Human input if interactive: `human_input`
- Existing instructions: `existing_common_instructions`, `existing_model_instructions`

Dynamic Parameters

- `max_common_instructions`, `max_model_specific_instructions`

Output Format (JSON):

```
{
  "updated_final_goal": {
    "goal_updated": "true|false",
    "sections_updated": ["..."],
    "updated_final_goal_text": "..."
  },
  "decision_mode_used": "human_interactive|gpt_autonomous",
  "conflict_resolutions": [
    {
      "conflict_id": "C1",
      "conflicting_candidates": ["Candidate A", "Candidate B"],
      "resolution_rationale": "Why this choice was made", ...
    }
  ],
  "finalized_common_instructions": [
    {
      "instruction_id": "CI1",
      "instruction_text": "Concrete common instruction",
      "addresses_patterns": ["P1", "P2"],
      "examples": [ ... ],
      "priority": "high|medium|low",
      "instruction_type": "new|improved|replacement", ...
    }
  ],
  "finalized_common_instructions_max": "{max_common_instructions}",
  "finalized_model_instructions": {
    "{model_name}": [ ... ]
  },
  "finalized_model_instructions_max_per_model":
    "{max_model_specific_instructions}",
  "instruction_generation_summary": ...
}
```

Table 10: Prompt for instruction refinement in phase 3 (Instruction Generation and Decision)

Prompt Text

You are a strict, methodical NER annotation supervisor acting as a guideline architect. Organize all instructions (existing + new) into a clear hierarchy, resolve remaining inconsistencies, and create the final guideline for the next iteration. Prioritize preservation of existing instructions and integrate new ones harmoniously.

Inputs - Phase 3 results: `phase3_results`
- Existing instructions: `existing_instructions`
- Original NER scheme: `current_ner_schema`
- Updated final task goal: `updated_final_goal`

Dynamic Parameters - `preserve_existing_instructions`, `limit_instruction_changes`, `max_change_ratio`

Output Format (JSON):

```
{
  "instruction_integration_analysis": {
    "existing_instructions_retained": ...
    "preservation_score": "0.0-1.0"
  },
  "contradiction_resolutions": [
    {
      "contradiction_id": "CR1",
      "conflicting_instructions": ["Instruction A", "Instruction B"], ...
    }
  ],
  "hierarchical_common_instructions": [
    {
      "level": "1",
      "instruction_number": "1",
      "instruction_text": "Top-level principle", ...
      "sub_instructions": [
        {
          "level": "1.1",
          "instruction_number": "1.1",
          "instruction_text": "Sub-principle",
          "examples": [
         {"text": "Example", "correct_annotation": "Gold", "explanation": "Note"}
          ], ... }]}],
  "prioritized_model_instructions": {
    "{model_name}": [
      {
        "priority_rank": 1,
        "instruction_id": "MI1_{model_name}",
        "instruction_text": "Highest-priority instruction", ...
      }]},
  "final_guideline_summary": {
    "total_hierarchical_common_instructions": 0,
    "max_hierarchy_depth": 2,
    ...
  }
}
```

Table 11: Prompt for instruction refinement in phase 4 (Hierarchical Guideline Organization)

Benchmark	Base	+dwa	$\Delta_{Base - dwa}$
AI	66.7	69.4	-2.7
Literature	72.0	73.7	-1.7
Music	80.6	80.1	+0.5
Politics	79.7	77.6	+2.1
Science*	74.8	74.8	+0.0
Movie	78.5	75.5	+3.0
Restaurant	62.1	67.8	-5.7
BroadTwitter	78.3	69.5	+8.8
CoNLL03	87.1	88.6	-1.5
FabNER	28.5	29.1	-0.6
MultiNERD	83.7	79.6	+4.1
ACE05	34.7	31.6	+3.1
AnatEM	60.6	55.6	+5.0
bc2gm	78.4	72.9	+5.5
bc4chemd	78.4	81.7	-3.3
bc5cdr	79.1	81.3	-2.2
GENIA	60.9	56.9	+4.0
OntoNotes	61.5	62.9	-1.4
Average	69.2	68.3	+0.9

Table 12: Zero-shot NER performance of DiZiNER with and without dropping the worst annotator (dwa). The worst model is defined as the model showing the highest average disagreement with all others. Bold indicates higher performance per benchmark.

Benchmark	Zero-shot	Supervised	$\Delta_{Sup. - ZS}$
AI	71.1	71.1	+0.0
Literature	73.8	72.9	-0.9
Music	82.9	79.9	-3.0
Politics	80.9	77.1	-3.8
Science	75.4	74.8	-0.6
Movie	78.5	75.0	-3.5
Restaurant	68.1	64.9	-3.2
BroadTwitter	78.3	75.2	-3.1
ACE05	46.2	55.8	+9.6
CoNLL2003	88.6	87.1	-1.5
MultiNERD	83.7	83.7	+0.0
OntoNotes	63.5	68.9	+5.4
FabNER	29.1	28.0	-1.1
AnatEM	60.7	60.1	-0.6
BC2GM	73.0	73.0	+0.0
BC4CHEMD	81.7	80.9	-0.8
BC5CDR	81.3	80.7	-0.6
GENIA	60.9	61.1	+0.2
Average	**72.4**	**71.4**	**-0.4**

Table 13: Comparison of DiZiNER zero-shot and supervised performance across benchmarks.