Is There a One-Model-Fits-All Approach to Information Extraction? Revisiting Task Definition Biases

Wenhao Huang[◇], **Qianyu He**[◇], **Zhixu Li**[◇], **Jiaqing Liang**[♡†], **Yanghua Xiao**[◇†]
[◇]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[♡]School of Data Science, Fudan University
{whhuang21,qyhe21}@m.fudan.edu.cn
{liangjiaqing,zhixuli,shawyh}@fudan.edu.cn

Abstract

Definition bias is a negative phenomenon that can mislead models. Definition bias in information extraction appears not only across datasets from different domains but also within datasets sharing the same domain. We identify two types of definition bias in IE: bias among information extraction datasets and bias between information extraction datasets and instruction tuning datasets. To systematically investigate definition bias, we conduct three probing experiments to quantitatively analyze it and discover the limitations of unified information extraction and large language models in solving definition bias. To mitigate definition bias in information extraction, we propose a multi-stage framework consisting of definition bias measurement, bias-aware fine-tuning, and task-specific bias mitigation. Experimental results demonstrate the effectiveness of our framework in addressing definition bias. [1].

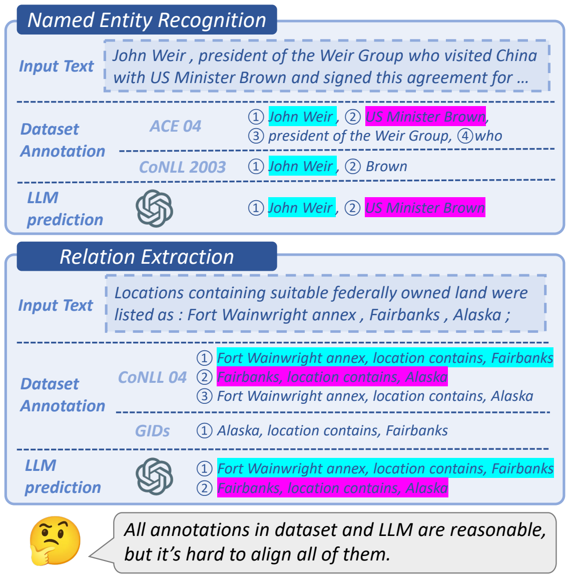

Figure 1: Definition bias among different datasets and LLMs even when they share the same entity type (for NER) or the same relation type (for RE).

1 Introduction

Bias in machine learning refers to systematic errors in predictions in the machine learning process, such as annotator bias, measurement bias, etc (Hellström et al., 2020). In the era of large language models (LLMs), these issues are addressed by filtering low-quality corpora (Kojima et al., 2022) and training with human preferences (Ouyang et al., 2022). However, performance remains subpar in handling information extraction (IE) tasks (Wadhwa et al., 2023), which we believe is due to definition bias.

Definition bias in IE refers to the tendency of an information extraction system to favour certain interpretations of data over others, often due to the way concepts, entities, or relationships are defined within the system. As the fast development of Unified Information Extraction (UIE) (Lu et al., 2022) and Large Language Models (LLMs) (OpenAI, 2022, 2023; Team et al., 2023) in recent years, two novel definition bias emerge, which are: *Bias among IE datasets* and *Bias between IE and instruction tuning (IFT) datasets*. Regarding *Bias among IE datasets*, it refers to the definition differences between different data sets under the same annotation schema. As illustrated in Figure 1, different datasets have different annotations to the same input for both Named Entity Recognition (NER) and Relation Extraction (RE) tasks. Regarding *Bias between IE and instruction tuning datasets*, it highlights the mismatch between the information extraction task and the general task. As depicted in Figure 1, although GPT-4 (OpenAI, 2023) is capable of extracting entities or relational triples in accordance with the specified task description without providing extra examples, its prediction differs from those in the existing datasets.

To systematically investigate definition bias in

[†]Corresponding authors.
[1]Resources of this paper can be found at https://github.com/EZ-hwh/definition-bias

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 10274–10287
November 12-16, 2024 ©2024 Association for Computational Linguistics

IE, we devise a series of probing experiments. First, we analyze *whether definition bias exists and how it varies among datasets sharing the same tasks*. By conducting cross-validation experiments among various datasets in the NER and RE, we observe a significant decrease in performance, indicating that definition bias negatively impacts the transferability of a fully-supervised model. An intuitive way to alleviate definition bias is unified information extraction, which is trained across multiple IE datasets. Therefore, we analyze *in the unified information extraction setting, does definition bias still exist?* By introducing source prompt (Li et al., 2022) that applies true or fake source names for the UIE models, we discover the inconsistency of the UIE for extraction, which indicates that UIE suffers from definition bias among IE datasets. The other way to mitigate definition bias is LLMs, which can understand a wide range of human instructions. Thereupon, we analyze *Can LLMs address the challenge of definition bias?* By conducting experiments on few-shot settings on NER and RE tasks with in-context learning, we find that it's difficult for LLMs without parameter updates to attain satisfactory performance, which indicates that LLMs still suffer from definition bias between IE and instruction fine-tuning datasets.

According to our probing experiments, it is imperative to address definition bias by proposing a universal solution for IE tasks. However, mitigating definition bias is non-trivial, primarily owing to the following three challenges. (1) Enhancing the capacity of LLMs in general information extraction tasks is vital to reduce the definition bias between information extraction datasets and instruction tuning datasets; (2) Mitigating the definition bias during the tuning of LLMs with different IE datasets; (3) Learning from new data over time, adapting to new tasks while ensuring the model remains good performance on existing tasks, is a significant challenge.

To address these challenges, we propose a framework to alleviate definition bias, which consists of definition bias measurement, bias-aware fine-tuning and task-specific bias mitigation. Using Fleiss's Kappa (Fleiss, 1971), we measure the two types of definition bias above. Then we conduct bias-aware fine-tuning with multiple information extraction instructions to enhance the extraction capabilities with less definition bias. Ultimately, we conduct the task-specific bias mitigation, with low-rank adaptation technique (LoRA) (Hu et al., 2021) for specific information extraction tasks to further align the LLMs with annotations.

Our paper is organized as follows: In Section 3, we present three probing experiments designed to explore the presence of definition bias and assess the ability of existing frameworks to address this issue in IE. Section 4 details the results and analysis of these experiments, concluding that frameworks based on either one-stage processing or parameter-free updates are insufficient to tackle definition bias in IE. Consequently, we propose a novel framework featuring two-stage fine-tuning, specifically developed to mitigate the identified definition bias, as introduced in Section 5. Ultimately, in Section 6, we compare the performance of our framework with state-of-the-art methods in universal information extraction, demonstrating its effectiveness in reducing definition bias.

2 Related Work

2.1 LLMs for information extraction

Large language models have shown remarkable performance in instruction following (OpenAI, 2023). To better align the natural instruction task from pre-trained and instruction tuning task, Wei et al. (2023); Wadhwa et al. (2023); Zhang et al. (2023) convert the structural information extraction task into natural instruction task such as question answering, multi-choice, etc. While Li et al. (2023); Guo et al. (2023) recast the structured output in the form of code to better leverage the LLMs of code to address the complex structure. Although LLMs show impressive performance in various information extraction tasks by designing fine-grained instruction, they still fail to address definition bias without further tuning.

2.2 Universal information extraction

Unified Information extraction, proposed by Lu et al. (2022), uniformly encodes various information extraction tasks with a predefined structured extraction language (SEL) and enhances the common IE abilities via a large-scale pre-trained generation model. Lou et al. (2023) further introduce USM to model different IE tasks, while Wang et al. (2023b) unified tasks into natural language instruction. GoLLIE converts the IE schema into the code-style structural description and adds guidelines to improve zero-shot results (Sainz et al., 2023). However, they mainly focus on how to encode different

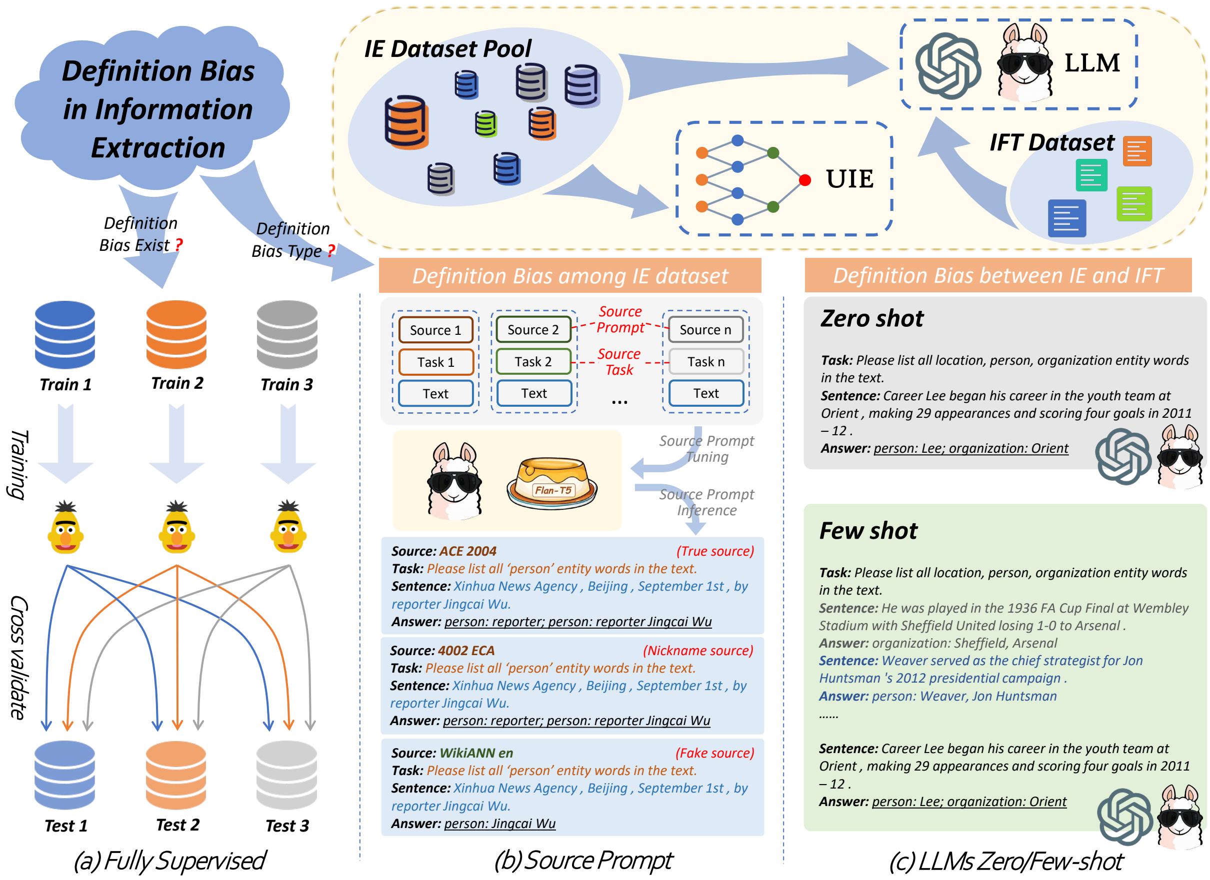

Figure 2: Three settings for the probing tasks on definition bias across datasets, including (a) fully supervised, (b) source prompt and (c) LLMs zero/few-shot.

extraction tasks into a uniform structure but fail to notice and detect the definition bias among various datasets.

3 Definition Bias Probing Experiment

We initially propose an experiment employing cross-validation to investigate the presence of definition bias in the IE tasks. Subsequently, we design two specific detection tasks: source prompt detection and few-shot prompting in LLMs, to examine two categories of definition bias: bias within IE datasets and bias between IE and instruction fine-tuning datasets. These experiments aim to explore the effectiveness of the UIE and LLM frameworks in addressing the definition bias issue.

3.1 Whether definition bias exists?

To better illustrate the definition bias among different information extraction tasks, we design a cross-extraction task. As shown in Figure 2(a), we train multiple fully-supervised models with different datasets on the same task (NER and RE) respectively, and test them on other datasets to evaluate whether definition bias exists.

We first introduce two BERT-based extraction frameworks to handle the NER and RE tasks, respectively.

Named Entity Recognition We adopt GlobalPointer (Su et al., 2022), an efficient span-based approach that models the beginning and end positions to predict entities using a two-dimensional scoring matrix. By incorporating extended softmax and cross-entropy, GlobalPointer is better equipped to learn from scenarios involving class imbalance.

Relation Extraction We adopt RERE (Xie et al., 2021) as the basic model for relation extraction. RERE is a pipeline approach that first performs sentence-level relation detection, followed by subject/object extraction. Specifically, the RERE model treats the former as a multi-class classification task and the latter as a span detection task.

During the cross-validation process, we encountered label-type biases across different datasets. For instance, the ACE 2004 dataset requires the extraction of the `weapon` entity, which is not a requirement in the CoNLL 2003 dataset. Consequently, we focus exclusively on the types of labels (such as entity types in NER and relationship types in RE) that are annotated in both the training and testing datasets. An example is the `person` label, which is common for both ACE 2004 and CoNLL 2003.

To mitigate the impact of text distribution shift on the experimental results, we sample a subset of sentences with similar semantics as a cross-validation set. Specifically, we measure the semantic similarity between two sentences by calculating the cosine similarity of their sentence embeddings. We define the semantic similarity of the sentence $sent_i$ to the dataset $\mathcal{D}$. Finally, we filter out all sentences that fall below $threshold(\mathcal{D})$.

$$sim(sent_i, \mathcal{D}) = \max_{ref_j \in \mathcal{D}} cosine(V_{sent_i}, V_{ref_j}) \quad (1)$$

$$threshold(\mathcal{D}) = \sigma \cdot \frac{1}{|\mathcal{D}|} \sum_{s_i \in \mathcal{D}} sim(s_i, \mathcal{D} \setminus \{s_i\}) \quad (2)$$

Where V_S denotes the embedding vector of a sentence S encoded by a sentence model[2], and σ denotes the hyperparameters that adjust the threshold, empirically set to 0.7.

3.2 Can UIE address definition bias?

Unified information extraction, which employs a pre-defined structured extraction language to encode different extraction structures, can accurately recognize extraction instructions. Inspired by Li et al. (2022), which introduces a novel prompt-based method in a transferable setting for text generation tasks, we adopt source prompt settings for probing. Briefly, in our experiment setting, a source can be denoted as the name of the dataset (e.g., ACE 2004). By presenting UIE with various sources—indicating which dataset the instance is from—we can guide it to yield different extraction results. This approach allows us to assess whether it can maintain consistent results with different source prompts.

As Figure 2(b) shows, the probing experiment consists of two parts: *source prompt tuning* and *source prompt inference*. Initially, we undertake a source prompt tuning process to enhance the UIE model's ability to recognize different sources. Subsequently, we examine the definition bias within the UIE model by introducing various sources.

[2]We adopt MPNet (Song et al., 2020) as our sentence embedding encoder, which is commonly used for retrieval.

Source Prompt Tuning The source prompt process can be regarded as a general multi-task learning framework. First, we define a set of source information extraction tasks $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_n\}$, where the k-th task $\mathcal{S}_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ contains N_k tuples of the input text $x_i^k \in \mathcal{X}_k$ and its corresponding output text $y_i^k \in \mathcal{Y}_k$. For a target information extraction task $\mathcal{T}$, the goal of multi-task learning is to leverage previously learned task-specific knowledge of the source tasks $\mathcal{S}$ to improve the prediction of the extraction result. Unlike the traditional multitask fine-tuning scenario, in source prompt tuning, we learn an independent *source prompt* p_k for each source information extraction task $\mathcal{S}_k$ in source prompt tuning, where x_i^k consists of extraction task source name s_k, information extraction task description t_k, and the sentence $sent_i^k$. For example, a single instance *"Here's a dataset from ACE 2004, please list all 'person' entity words in the text. Input sentence: Xinhua News Agency, Beijing, September 1st, by reporter Jingcai Wu."* contains the components that are described above.

To demonstrate that UIE with instruction tuning can implicitly learn the definitions of a dataset through source prompt, we assign a nickname p'_k for every dataset and randomly replace p_k with p'_k. For simplicity, we merely reverse the order of the original dataset names, thereby generating non-natural language nicknames. For example, the dataset name *"ACE 2004"* is replaced with *"4002 ECA"*. This procedure is designed to eliminate the influence caused by the differences in learning various source names in the UIE and to ensure that the discrepancies in results between true and fake settings are solely due to dataset definition bias.

Specifically, we adopt Llama-v2-13B (Touvron et al., 2023) and FlanT5-11B (Chung et al., 2022) as our backbone models in source prompt tuning settings because of their powerful instruction understanding and instruction-following capabilities. Based on multiple datasets in NER and RE, we add an additional *source prompt* to every extraction instance to indicate the dataset to which it belongs. Further details on source prompt tuning are described in the Appendix A.2.

Source Prompt Inference In reference, we provide different source prompts with the same extraction instance to our UIE models that have been fine-tuned on the dataset with source prompts. To probe the definition bias in universal generative information extraction, UIE predicts the extraction result with ***True source*** (the extraction case with the original source name), ***Nickname source*** (a nickname of the original source name) and ***Fake source*** (the extraction case with a fake source name). With different source names, UIE generates different extraction results following different definitions learned from source prompt tuning.

3.3 Can LLMs address definition bias?

Large language models exhibit remarkable instruction understanding capabilities, which help them achieve extraordinary performance on various tasks. However, due to the definition bias between IE datasets and IFT datasets, there is a significant performance gap in LLMs when it comes to the information extraction task (Wadhwa et al., 2023). In-context learning, where LLMs make predictions based solely on contexts augmented with a few examples, is a training-free learning framework that enables models to adapt to new tasks (Dong et al., 2023). It is considered a solution to address the definition bias between IE datasets and instruction tuning datasets.

As shown in Figure 2(c), we conduct the probing experiment with multiple LLMs in both zero-shot and few-shot settings.

Specifically, we utilize open-source LLMs such as Llama-v2-chat-70B (Touvron et al., 2023), and close-source LLMs GPT-3.5-Turbo (OpenAI, 2022), GPT-4 (OpenAI, 2023) as our backbone models. In zero-shot settings, we prompt LLMs with a task description, which probes the definition bias between IE and IFT datasets. Meanwhile, in few-shot settings, we prompt LLMs with a task description and an additional four cases randomly sampled from the corresponding training set to examine whether in-context learning can address the definition bias. For a fair comparison, we sample 200 cases from each dataset and test them in both zero-shot and few-shot settings, respectively.

4 Empirical Study of Definition Bias

4.1 Whether definition bias exists?

Following the cross-validation setting described in Section 3.1, we experiment separately on NER and RE tasks in the general domain. Table 1,2 show the validation result in fully-supervised settings.

Briefly, we define the model trained and tested on the same dataset as the *reference model*. The numbers in the table cells represent the F1 scores when compared to the golden label. Additionally,

	A04[1]	**A05**[2]	**C03**[3]	**Ont**[4]	**Wie**[5]	**TN7**[6]	**WiN**[7]	**PoN**[8]
A04[1]	85.10	82.19	35.77	28.89	49.89	28.06	30.54	17.64
A05[2]	83.44	84.45	37.80	26.43	46.53	26.94	29.09	18.23
C03[3]	24.10	16.57	92.19	55.82	55.10	78.26	92.08	53.67
Ont[4]	32.53	21.20	60.60	89.69	49.76	34.75	61.23	37.58
Wie[5]	23.09	8.42	67.10	41.14	86.60	61.99	70.96	44.13
TN7[6]	25.60	21.07	76.16	56.15	73.95	63.39	82.70	54.45
WiN[7]	25.48	20.61	80.10	58.69	57.33	63.44	95.21	51.96
PoN[8]	14.58	10.84	44.36	35.28	40.26	32.65	69.66	77.77

[1] ACE 2004 [3] CoNLL 2003 [5] WikiANN en [7] WikiNeural
[2] ACE 2005 [4] Ontonotes [6] TweetNER 7 [8] PolyglotNER

Table 1: Definition bias among different NER tasks.

	CoNLL 04	**NYT10**	**NYT11**	**GIDs**	**WikiKBP**
CoNLL 04	61.12	10.20	12.07	-	26.98
NYT10	14.36	89.68	52.29	14.33	30.32
NYT11	8.78	83.32	56.82	10.70	32.64
GIDs	-	7.77	6.45	65.12	55.65
WikiKBP	0.00	15.05	2.53	26.49	36.57

Table 2: Definition bias among different RE tasks. Cells with (-) indicates that there are no same relation types between the two datasets.

the depth of colour in each cell indicates the relative quality of extraction in comparison to the reference model. In other words, the darker the cell colour, the closer the extraction results are to those of the reference model. The rows of the table represent the training dataset, while the columns represent the test dataset.

Intuitively, the deepest red cells are distributed along the diagonal of the entire table, illustrating that definition bias exists among different datasets, even though they share the same types. This is particularly evident in NER tasks, where several datasets focus on common entity types such as `person`, `location`, and `date`. Despite these similarities, definition bias can lead to significant variations in the model's extraction capabilities.

4.2 Can UIE address definition bias?

Following the source prompt setting described in Section 3.2, we tuned Llama-13b and Flan-T5 with source prompt instructions and prompted them with three source settings.

Table 3 displays the extraction results evaluated by F1 scores. Replacing the true source names with fake ones results in a drop in F1 scores across all NER and RE tasks, with an average decrease of

Model	Llama-13b			Flan-T5		
Source	*True*	*Nickname*	*Fake*	*True*	*Nickname*	*Fake*
			Named Entity Recognition			
ACE 04	84.93	84.89	60.85	77.82	78.41	45.79
ACE 05	84.85	85.16	61.56	79.20	79.59	44.10
CoNLL 03	81.02	80.87	73.34	78.94	78.84	69.23
Ontonotes	91.85	91.81	81.79	91.03	91.04	78.71
WikiANN en	89.54	89.65	81.43	76.26	76.07	66.08
TweetNER 7	68.92	69.11	66.19	68.35	68.45	60.44
WikiNeural	96.03	95.93	83.51	94.03	94.03	74.30
PolyglotNER	80.21	80.41	68.34	74.00	74.03	54.24
avg	-	-	-12.6	-	-	-18.4
			Relation Extraction			
CoNLL 04	69.88	69.51	61.73	67.09	67.00	57.34
NYT10	97.80	97.78	94.82	96.20	96.20	90.54
NYT11	76.14	76.24	72.82	76.14	76.41	71.94
GIDs	80.49	80.15	78.69	76.41	76.34	74.26
WikiKBP	64.68	65.67	63.50	63.78	63.94	59.64
avg	-	-	-3.5	-	-	-5.2

Table 3: Different extraction results obtained by prompting the source prompt tuning UIE with *true*, *nickname* and *fake* source name.

Dataset	Llama-chat-70B	GPT-3.5-Turbo	GPT-4
ACE04	8.56 | 30.42	19.68 | 32.81	13.70 | **35.16**
ACE05	17.64 | 33.48	20.83 | 34.32	16.13 | **45.30**
CoNLL 03	33.89 | 49.36	39.70 | 55.90	46.66 | **64.99**
Ontonotes	11.86 | 27.56	22.14 | 28.83	31.70 | **40.57**
WikiANN en	32.87 | 50.00	50.83 | 57.90	51.57 | **59.03**
TweetNER 7	31.77 | 35.68	32.98 | 38.13	36.62 | **47.88**
WikiNeural	42.98 | 57.03	50.00 | 59.83	65.23 | **70.66**
PolyglotNER	21.44 | 30.91	42.20 | 44.88	**45.14** | 43.23
CoNLL 04	3.36 | 18.77	9.22 | 23.86	24.62 | **29.86**
NYT10	2.97 | 13.17	2.13 | 13.64	16.67 | **20.13**
NYT11	2.03 | 5.33	1.93 | 6.50	8.00 | **12.00**
GIDs	11.36 | 7.92	7.89 | 19.45	6.82 | **24.54**
WikiKBP	18.55 | 29.56	17.25 | 32.41	25.00 | **45.85**

Table 4: Performance of Open-source LLM and closed-source LLM on various information extraction tasks in (**zero-shot | few-shot**) settings.

12.6/3.5 and 18.4/5.2, respectively. However, when true source names are replaced with nicknames, the results show virtually no difference. This significant performance gap highlights that UIE is unable to mitigate definition bias during the multi-task learning process. The implicit definition bias permeates the model, leading to inconsistent extraction results, even when the same extraction task instructions are given.

4.3 Can LLM address definition bias?

The performance of various models on different tasks is presented in Table 4. Among the evaluated models, GPT-4 stands out by achieving the best performance across almost all datasets in both zero-shot and few-shot settings. Furthermore, the few-shot settings, which incorporate similar cases from the same dataset into the context, enhance performance by an average of 9.82 compared to the zero-shot settings. This improvement underscores the capacity of in-context learning to partially mitigate definition bias.

Despite these advances, it remains challenging for conventional, off-the-shelf methods to reach the performance levels of fully supervised approaches. This discrepancy underscores the presence of a significant definition bias between datasets used for information extraction and those used for instruction fine-tuning. Additionally, applying LLMs to information extraction faces two primary limitations. First, the constraint on context length prevents the inclusion of all annotated cases within the context. Second, the definition bias across different information extraction datasets complicates the creation of comprehensive prompts that accurately describe the extraction tasks.

5 Alleviate Definition Bias

In this section, we explore methods to enhance the information extraction capabilities of LLMs.

Based on the probing experiments and conclusions outlined in Sections 3 and 4, we find that definition bias across different datasets significantly impacts the performance of UIE and LLMs. This indicates that a framework relying solely on a one-stage, parameter-free update is inadequate for addressing definition bias. To tackle this challenge, we introduce a two-stage fine-tuning framework. Moreover, by identifying and explicitly quantifying the two types of definition biases we have discovered, we can integrate these measurements into our fine-tuning framework, effectively reducing the influence of definition bias.

5.1 Definition bias measurement

First, we introduce **Fleiss' Kappa**, a statistical measure used to assess the reliability of agreement among multiple raters when they assign categorical ratings to a set of items. This tool is valuable in identifying and mitigating definition bias.

$$\kappa = 1 - \frac{1 - p_0}{1 - p_e} = \frac{p_0 - p_e}{1 - p_e} \tag{3}$$

Where p_o denotes the *Observed Agreement*, the proportion of times that the raters actually agree, and p_e denotes the *Expected Agreement*, which

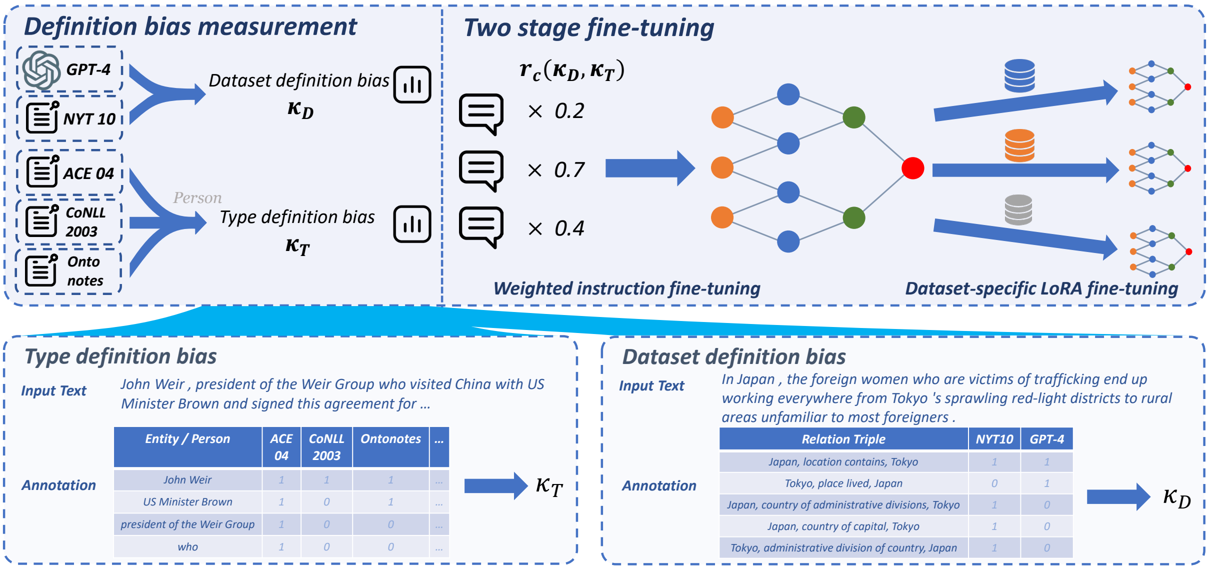

Figure 3: Our two-stage framework for alleviating definition bias. **Left:** we measure two kinds of definition bias with Fleiss' Kappa; **Right:** we first full-parameter fine-tune LLMs with measurement and then fine-tune with LoRA on a specific dataset.

represents the agreement that could be expected purely by chance. Suppose there are N cases for a task, and each data is labelled n times, and k is the number of categories. These can be calculated using the following formula.

$$p_e = \sum_{j=1}^{k} p_j^2, \quad p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij} \tag{4}$$

$$p_o = \frac{1}{N} \sum_{i=1}^{N} p_i, \quad p_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1) \tag{5}$$

where n_{ij} denotes the number of annotator that label case i as category j.

Specifically, we focus on the definition bias in information extraction and divide the definition bias into two types: ***dataset definition bias*** and ***type definition bias***.

Dataset Definition Bias κ_D recognized as the agreement between GPT4 and the annotation of the dataset, serving as a measure of reliability for transforming information extraction into instruction tuning dataset. It is carried out by calculating Fleiss's Kappa between the GPT4 extraction results and the golden annotation of the dataset.

Type Definition Bias κ_T considered as the agreement among information extraction datasets with the same type either entity or relationship, and serves as a metric to evaluate the reliability of these types in terms of consistent annotation.

5.2 Bias-aware fine-tuning

Based on the probing experiments, inconsistencies in definitions across various datasets significantly impact the training process. However, the diversity among these datasets, including annotation types and text sources, helps to improve the performance of LLMs for IE tasks. Therefore, it is essential to adopt a precise method to assess the quality of different datasets to effectively guide the training process.

We fine-tune the LLMs with information extraction dataset through C-RLFT (Wang et al., 2023a), which enables leveraging mixed-quality training data. We define the quality of the training samples as metrics based on κ_D and κ_T. Suppose there are N entity or relation triples in a case, we calculate the coarse-grained rewards of each case $r_c(x_i, y_i)$ by the formula below.

$$r_c(x_i, y_i) = (1 + \kappa_D) \frac{1}{N} \sum_{i=1}^{N} \kappa_{T_i} \tag{6}$$

5.3 Task-specific bias mitigation

To further enhance the performance of LLMs on specific information extraction datasets, we employ Low-Rank Adaptation (LoRA) for additional instruction tuning. We hypothesize that updates to the weights for a dataset possess a low intrinsic rank. This low intrinsic dimension adaptation can help mitigate the definition bias between a multi-

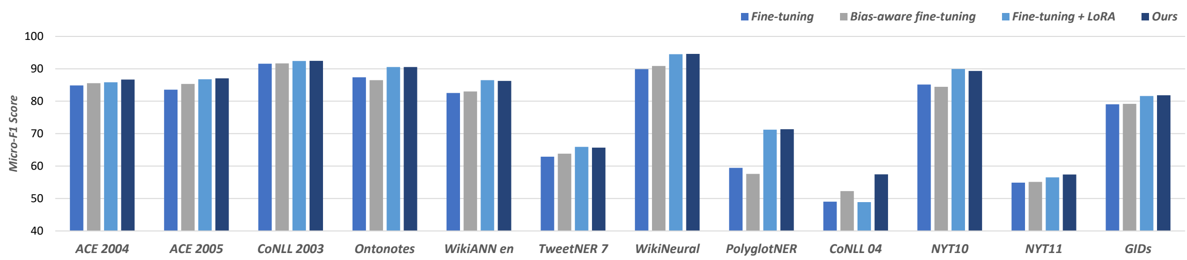

Figure 4: Ablation study on 12 information extraction datasets (NER and RE)

Dataset	UIE	USM	InstructUIE	Ours
ACE 04	86.89	**87.62**	-	86.68
ACE 05	85.78	**87.14**	86.66	87.05
CoNLL 03	92.99	**93.16**	92.94	92.47
Ontonotes	-	-	90.19	**90.52**
WikiANN en	-	-	85.13	**86.24**
TweetNER 7	-	-	64.97	**65.70**
WikiNeural	-	-	91.36	**94.59**
PolyglotNER	-	-	70.15	**71.34**
CoNLL 04	75.00	**78.84**	78.48	57.46
NYT10	-	-	**90.47**	89.35
NYT11	-	-	56.06	**57.38**
GIDs	-	-	**81.98**	81.83

Table 5: Main result for comparing with other models on NER and RE tasks.

task learning model and the dataset. Specifically, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, we constrain its update with a low-rank decomposition.

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{7}$$

where $\Delta W x$ denotes the updatable parameters of W_0, and it can be constrained with a low-rank decomposition $\Delta W = BA$, where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and the rank $r \ll min(d, r)$. W_0 is frozen and does not receive gradient updates while A and B contain trainable parameters during training.

In this stage, the model updates its parameter through further fine-tuning on a specific dataset to align the annotation.

6 Experiments of Two-stage Framework

This section conducts experiments to validate the effectiveness of our two-stage fine-tuning framework. We select 11B Flan-T5 (Chung et al., 2022) as our backbone model. The details of the experimental setup and comparison methods are described in the following parts.

6.1 Experimental setup

In bias-aware fine-tuning, we apply a sampling strategy to balance the dataset. In specific, we sample 10,000 cases from each dataset for training. In task-specific bias mitigation, we apply all examples for training. Further details can be found in Appendix B.2.

Our baseline models contain: **UIE** (Lu et al., 2022), **USM** (Lou et al., 2023), and **InstructUIE** (Wang et al., 2023b).

6.2 Results

Table 5 presents the result on different datasets with baselines. Although our framework was trained on several information extraction datasets in the general domain, which might be considered unfair for comparing with baselines trained on other datasets, it achieves state-of-the-art on many datasets. It is worth noting that in datasets focusing exclusively on `person`, `location`, `organization` (as listed in Table 16), our framework achieves the best performance on WikiANN en, WikiNeural and PolyglotNER. This demonstrates the effectiveness of our framework in mitigating definition bias across different datasets.

6.3 Experiment with two-stage fine-tuning

To better improve the effectiveness of our two-stage fine-tuning framework, we conduct an ablation study comparing with the following baseline: 1. ***Fine-tuning***: fine-tuning the model with information extraction; 2. ***Bias-aware fine-tuning***: first stage fine-tuning in Section 5.2; 3. ***Fine-tuning+LoRA***: data-specific instruction-tuning with LoRA on the weight of baseline 1; 4. ***Ours***: our two-stage fine-tuning framework.

The results are shown in Figure 4. In general, our framework nearly achieves the best performance compared to the baseline, demonstrating its effectiveness. By comparing baselines 1 and 2, it is proven that our bias-aware fine-tuning can alleviate definition bias among IE datasets and help models better align with GPT-4. It is also notable that two-stage fine-tuning consistently improves performance on specific datasets, attributed to task-specific bias mitigation.

7 Conclusion

In the paper, we propose the definition bias problem in information extraction tasks. We conduct several probing experiments to comprehensively demonstrate that existing methods cannot address definition bias. We then propose a multi-stage tuning framework, which consists of bias-aware fine-tuning and task-specific bias mitigation, to alleviate the definition bias in a specific dataset. Experimental results show that our framework is efficient in mitigating definition bias.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 62102095). The computations in this research were performed using the CFFF platform of Fudan University.

Limitation

We systematically investigate definition bias in IE with devising a series of probing experiments. And we propose a multi-stage framework to mitigate definition bias in IE. However, there are still some limits of our probing experiment and the solution framework.

First, our probing experiment only focus on the definition bias among NER and RE tasks, which does not cover all the task in information extraction, which remains improvement for the future work.

Second, the performance of our solution framework is restricted by two main reason: 1) more diverse dataset can be used for the bias-aware fine-tuning dataset; 2) the choice on backbone model also plays an important role in model performance. More experiments can more effectively validate the effectiveness of the proposed framework.

Ethic statement

We hereby declare that all authors of this article are aware of and adhere to the provided ACL Code of Ethics and honor the code of conduct.

Use of Human Annotations Human annotations are only utilized in the early stages of methodological research to assess the feasibility of the proposed solution. All annotators have provided consent for the use of their data for research purposes. We guarantee the security of all annotators throughout the annotation process, and they are justly remunerated according to local standards. Human annotations are not employed during the evaluation of our method.

Risks The datasets used in the paper have been obtained from public sources and anonymized to protect against any offensive information. Though we have taken measures to do so, we cannot guarantee that the datasets do not contain any socially harmful or toxic language.

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Joe Ellis, Xuansong Li, Kira Griffitt, Stephanie M Strassel, and Jonathan Wright. 2012. Linguistic resources for 2013 knowledge base population evaluations. In *TAC*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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

Thomas Hellström, Virginia Dignum, and Suna Bensch. 2020. Bias in machine learning – what is it good for?

Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. 2018. Improving distantly supervised relation extraction using word and entity based attention. *arXiv preprint arXiv:1804.06987*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Junyi Li, Tianyi Tang, Jian-Yun Nie, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Learning to transfer prompts for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3506–3518.

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*.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching.

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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2005. Ace 2004 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 1:1–1.

OpenAI. 2022. Chatgpt.

OpenAI. 2023. Gpt-4 technical report.

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

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21*, pages 148–163. Springer.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the eighth conference on computational natural language learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8.

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.

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

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global pointer: Novel efficient span-based approach for named entity recognition.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7072–7079.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, and Anja Hauth. 2023. Gemini: A family of highly capable multimodal models.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Asahi Ushio, Leonardo Neves, Vitor Silva, Francesco Barbieri, and Jose Camacho-Collados. 2022. Named entity recognition in twitter: A dataset and analysis on short-term temporal shifts. *arXiv preprint arXiv:2210.03797*.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. *arXiv preprint arXiv:2305.05003*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

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. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Chenhao Xie, Jiaqing Liang, Jingping Liu, Chengsong Huang, Wenhao Huang, and Yanghua Xiao. 2021. Revisiting the negative data of distantly supervised relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3572–3581, Online. Association for Computational Linguistics.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. *arXiv preprint arXiv:2305.11159*.

	A04[1]	**A05**[2]	**C03**[3]	**Ont**[4]	**Wie**[5]	**TN7**[6]	**WiN**[7]	**PoN**[8]
A04[1]		451	1299	3404	1886	121	2104	1260
A05[2]	489		1112	4649	3552	204	1954	1752
C03[3]	71	67		409	2576	22	1128	550
Ont[4]	569	770	1149		4194	180	2880	2504
Wie[5]	248	310	2338	2086		279	7752	6957
TN7[6]	222	328	1486	1736	3016		2397	1648
WiN[7]	387	407	2239	2843	8230	229		7624
PoN[8]	558	669	2795	4918	9597	387	10869	
#Tot	812	1060	3453	8262	10000	576	11597	10000

[1] ACE 2004
[2] ACE 2005
[3] CoNLL 2003
[4] Ontonotes
[5] WikiANN en
[6] TweetNER 7
[7] WikiNeural
[8] PolyglotNER

Table 6: Filter result in NER task.

	CoNLL 04	**NYT10**	**NYT11**	**GIDs**	**WikiKBP**
CoNLL 04		1935	150	1879	72
NYT10	242		344	3977	160
NYT11	189	4032		3608	150
GIDs	52	887	83		79
WikiKBP	3	71	8	70	
#Tot	288	5000	369	4307	182

Table 7: Filter result in NER task.

A Further Details about Probing Experiments

In this section, we further introduce the details of three probing experiments, including experiment settings and additional experiment results.

A.1 Fully supervised settings

In fully supervised settings, we trained GlobalPointer (Su et al., 2022) and ReRe (Xie et al., 2021) for each task in 20 epochs, with a learning rate of 2e-5 and a batch size of 32.

Table 6 and Table 7 show the filter result with a similar semantic filtering process. While the cross-validation result without the semantic similarity filter process is demonstrated in Table 8 and 9.

A.2 Source prompt settings

To eliminate the instruction bias that different datasets focus on different types of entity or relation, we employ a task decomposition approach, which involves constructing separate instructions for every entity type or relationship type. It helps decompose a task instruction with many types into atomic task instructions, which can be shared across different datasets. Such a setting compels the UIE model to focus solely on the source name

	A04[1]	A05[2]	C03[3]	Ont[4]	Wie[5]	TN7[6]	WiN[7]	PoN[8]
A04[1]	85.10	81.54	39.36	31.15	41.70	35.65	29.70	21.10
A05[2]	82.87	84.45	40.31	30.05	40.25	34.72	29.18	20.61
C03[3]	25.63	19.53	92.19	61.11	55.06	68.26	88.89	51.48
Ont[4]	35.04	25.89	59.49	89.69	45.16	37.88	62.20	37.42
Wie[5]	19.18	14.11	64.56	39.22	86.60	59.52	64.52	39.28
TN7[6]	24.78	19.10	71.61	46.84	58.62	63.39	74.17	47.04
WiN[7]	24.10	18.77	78.76	58.69	55.41	64.51	95.21	51.36
PoN[8]	14.49	10.17	46.30	38.18	40.26	34.45	69.45	77.77

[1] ACE 2004 [2] ACE 2005 [3] CoNLL 2003 [4] Ontonotes [5] WikiANN en [6] TweetNER 7 [7] WikiNeural [8] PolyglotNER

Table 8: Definition bias between different NER tasks without similar semantic filtering.

	CoNLL 04	NYT10	NYT11	GIDs	WikiKBP
CoNLL 04	61.12	9.17	7.54	-	24.37
NYT10	14.36	89.68	53.16	15.17	30.30
NYT11	8.78	83.13	56.82	12.50	31.40
GIDs	-	7.44	3.17	65.12	51.67
WikiKBP	0.00	7.95	2.53	18.78	36.57

Table 9: Definition bias between different RE tasks without similar semantic filtering.

and apply the distinct extraction principle for different source prompts. Table 10 shows a case of task decomposition on NER.

A.3 Large language model zero/few-shot settings

We use the prompt in Table 11 for probing experiments and multi-stage fine-tuning, which consists of the task description, output format, in-context learning cases and input text.

B Further Details about Two-stage Instruction Fine-tuning

B.1 Fleiss' Kappa

Table 12 and 13 show the κ_T and κ_D that measure the ***type definition bias*** and ***dataset definition bias*** in several IE datasets.

B.2 Two-stage training settings

The training hyperparameters of our multi-stage framework are listed in Table 14.

C Detail Statistic on Training Datasets

We use 13 datasets in named entity recognition and relation extraction. For NER task, the dataset include ACE04 (Mitchell et al., 2005), ACE05 (Walker et al., 2006), CoNLL2003 (Sang and De Meulder, 2003), Ontonotes (Hovy et al., 2006), PolyglotNER (Al-Rfou et al., 2015), TweetNER (Ushio et al., 2022), WikiNeural (Tedeschi et al., 2021) and WikiANN (Pan et al., 2017). For RE task, the datasets include CoNLL 2004 (Roth and Yih, 2004), GIDS (Jat et al., 2018), NYT10 (Riedel et al., 2010), NYT11-HRL (Takanobu et al., 2019) and Wiki-KBP (Ellis et al., 2012).

The statistics of datasets are listed in Table 15. The pre-defined entity or relation types for each dataset are shown in Table 16.

Original Extraction Instruction
Instruction: Please list all entity words in the text that fit the category. Here's the category list: [person,organization,location] And then output the result in the format of "'type1: entity1; type2: entity2; ...'' Input: *[Input text for NER]* Output:
Decomposed Extraction Instruction
Instruction: Please list all entity words in the text that fit the category. Here's the category list: [person] And then output the result in the format of "'type1: entity1; type2: entity2; ...'' Input: *[Input text for NER]* Output:
Instruction: Please list all entity words in the text that fit the category. Here's the category list: [organization] And then output the result in the format of "'type1: entity1; type2: entity2; ...'' */*Input text*/* Input: *[Input text for NER]* Output:
Instruction: Please list all entity words in the text that fit the category. Here's the category list: [location] And then output the result in the format of "'type1: entity1; type2: entity2; ...'' Input: *[Input text for NER]* Output:

Table 10: A case for decomposing NER tasks instruction which focuses on the entity type: `person`, `organization` and `location`.

Prompt for Named Entity Recognition

*/*Task prompt*/*

Instruction: Please list all entity words in the text that fit the category. Here's the category list:

*/*Entity type List*/*

[List of the entity type]

*/*Output Format*/*

And then output the result in the format of "'type1: entity1; type2: entity2; ...''

*/*In-context learning cases*/*

*/*Input text*/*

Input: *[Input text for NER]*

Output:

Prompt for Relation Extraction

*/*Task prompt*/*

Instruction: Given a sentence or paragraph, and a given relationship set that describe the relation between entities. Here's the relation set:

*/*Relation type List*/*

[List of the relationship type]

*/*Output Format*/*

Output the result in the format of "'(subject1, relation1, object1), (subject2, relation2, object2), ...''

*/*In-context learning cases*/*

*/*Input text*/*

Input: *[Input text for RE]*

Output:

Table 11: Prompts for two type of information extraction task: NER and RE.

Type	Fleiss' Kappa
Entity Type of Named Entity Recognition	
person	0.414
location	0.428
organization	0.364
facility	0.021
Relation Type of Relation Extraction	
place lived	0.473
place of birth	0.467
place of death	0.408
children	0.333
location contains	0.150
person of company	0.359

Table 12: κ_T measured with dataset annotation

Dataset	Fleiss' Kappa
ACE 2004	-0.648
ACE 2005	-0.546
CoNLL 2003	-0.350
Ontonotes	-0.594
PolyglotNER	-0.567
TweetNER7	-0.521
WikiANN en	-0.409
WikiNeural	-0.293
conll04	-0.701
GIDS	-0.748
NYT10	-0.799
NYT11	-0.879
WikiKBP	-0.541

Table 13: κ_D measured with dataset annotation and GPT-4

Hyperparameters	Settings
Bias-aware fine-tuning	
Learning rate	1e-5
Epoch	5
Batch size	384
Dataset-specific Mitigation	
Learning rate	1e-5
LoRA rank	8
LoRA_key	q,v
Epoch	10/30
Batch size	256

Table 14: Hyper-parameters for two-stage training with Flan-T5.

Dataset	#Train	#Valid	#Test
Named Entity Recognition			
ACE 2004	6202	745	812
ACE 2005	7299	971	1060
CoNLL 03	14041	3250	3453
Ontonotes	59924	8528	8262
PolyglotNER	393982	10000	10000
TweetNER 7	7111	886	576
WikiANN en	20000	10000	10000
WikiNeural	92720	11590	11597
Relation Extraction			
CoNLL 04	922	231	288
GIDs	8526	1417	4307
NYT10	56196	5000	5000
NYT11	62648	149	369
WikiKBP	79934	20	182

Table 15: Detailed datasets statistic.

Dataset	Annotation type
	Named Entity Recognition
ACE 2004	geographical social political, organization, person, location, facility, vehicle, weapon
ACE 2005	organization, person, geographical social political, vehicle, location, weapon, facility
CoNLL 03	location, else, organization, person
Ontonotes	date, organization, person, geographical social political, national religious political, facility, cardinal, location, work of art, law, event, product, ordinal, percent, time, quantity, money, language
PolyglotNER	location, person, organization
TweetNER 7	group, creative work, person, event, product, location, corporation
WikiANN en	location, person, organization
WikiNeural	location, person, organization
	Relation Extraction
CoNLL 04	company founded place, location contains, place lived, person of company, kill
GIDs	place of death, place of birth, education degree, education institution
NYT10	ethnicity, place lived, geographic distribution, company industry, country of administrative divisions, administrative division of country, location contains, person of company, profession, ethnicity of people, company shareholder among major shareholders, sports team of location, religion, neighborhood of, company major shareholders, place of death, nationality, children, company founders, company founded place, country of capital, company advisors, sports team location of teams, place of birth
NYT11	nationality, country capital, place of death, children, location contains, place of birt, place lived, administrative division of country, country of administrative divisions, company, neighborhood of, company founders
WikiKBP	parent, children, person of company, place of birth, place of death, place lived, religion

Table 16: The type of entity or relationship in each dataset.