ADELIE: Aligning Large Language Models on Information Extraction

Anonymous ACL submission

Abstract

Large language models (LLMs) usually fall short on information extraction (IE) tasks and struggle to follow the complex instructions of IE tasks. This primarily arises from LLMs not being aligned with humans, as mainstream alignment datasets typically do not include IE data. In this paper, we introduce **ADELIE** (**A**ligning large language mo**DEL**s on **I**nformation **E**xtraction), an aligned LLM that effectively solves various IE tasks, including closed IE, open IE, and on-demand IE. We first collect and construct a high-quality alignment corpus `IEInstruct` for IE. Then we train ADELIE$_{\text{SFT}}$ using instruction tuning on `IEInstruct`. We further train ADELIE$_{\text{SFT}}$ with direct preference optimization (DPO) objective, resulting in ADELIE$_{\text{DPO}}$. Extensive experiments on various held-out IE datasets demonstrate that our models (ADELIE$_{\text{SFT}}$ and ADELIE$_{\text{DPO}}$) achieve state-of-the-art (SoTA) performance among open-source models. We further explore the general capabilities of ADELIE, and experimental results reveal that their general capabilities do not exhibit a noticeable decline. We will release the code, data, and models to facilitate further research.

1 Introduction

Large language models (LLMs), especially after alignment with human expectations, such as instruction tuning (Wei et al., 2022a; Chung et al., 2022; Longpre et al., 2023) or direct preference optimization (DPO) (Rafailov et al., 2023), have achieved impressive results on numerous tasks (OpenAI, 2022, 2023; Jiang et al., 2023; Anil et al., 2023; Anthropic, 2024). However, LLMs still fall short on information extraction (IE) tasks, particularly on closed IE tasks (Li et al., 2023a; Han et al., 2023; Peng et al., 2023a). LLMs usually struggle to understand and follow the complex instructions of IE tasks (Peng et al., 2023a; Pang et al., 2023; Xu et al., 2023), e.g., complicated task schema and specifications, which indicates existing LLMs are not aligned with human needs on IE tasks (Peng et al., 2023a; Sainz et al., 2023).

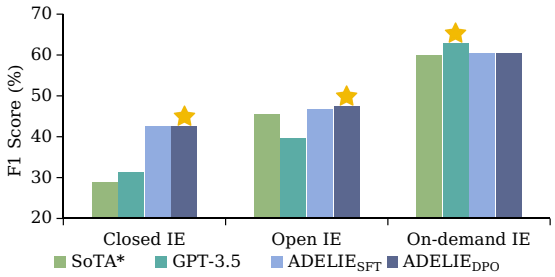

Figure 1: F1 scores (%) on closed, open, and on-demand IE tasks in the few-shot setting. SoTA* denotes the best performance of open-source models.

To enhance LLM performance on IE tasks, existing efforts have primarily explored three aspects: (1) **Prompt engineering**, which provides comprehensive information, e.g., annotation guidelines, to LLMs, without fine-tuning model parameters (Pang et al., 2023; Guo et al., 2023; Wei et al., 2023b; Wan et al., 2023). (2) **Code LLMs**, which leverage their capabilities of understanding structured information to enhance the performance on IE tasks (Guo et al., 2023; Sainz et al., 2023; Bi et al., 2023). (3) **Multi-task fine-tuning**, which involves fine-tuning LLMs on multiple IE datasets to enhance their cross-task generalization capabilities in solving IE tasks (Wang et al., 2022a, 2023b; Sainz et al., 2023; Wang et al., 2023d).

However, these works do not sufficiently align LLMs on IE tasks. The prompt engineering method does not inherently align LLMs without tuning model parameters. Works using code LLMs and multi-task fine-tuning typically fine-tune models on homogeneous data, e.g., instances with the same input-output format, with a lack of diverse alignment data. Therefore, the fine-tuned models exhibit limited generalization capabilities on IE tasks, in-

cluding closed IE (Xu et al., 2023), open IE (Xu et al., 2023), and on-demand IE (Jiao et al., 2023). Furthermore, as these models are trained specifically for IE, their general capabilities, such as natural language understanding (Hendrycks et al., 2021), may experience a significant decline.

Considering the above issues, we introduce **ADELIE** (**A**ligning large language mo**DEL**s on **I**nformation **E**xtraction), an LLM aligned on IE tasks. Specifically, this work addresses the above limitations through two aspects: (1) **Rich alignment data**. We construct a high-quality instruction tuning dataset for IE tasks, `IEInstruct`, including $83,585$ instances of various IE tasks. `IEInstruct` includes a diverse set of instructions and input-output formats. We manually write several instructions for different IE tasks, then expand the instruction set using GPT-4 (OpenAI, 2023) similar to Self-Instruct (Wang et al., 2023e). We then augment the instructions through various augmentation techniques, such as adding annotation guidelines (Sainz et al., 2023). `IEInstruct` also includes diverse output formats, such as triplets, natural language, and JSON. We also employ GPT-4 to generate chain-of-thought explanations (Wei et al., 2022c) for $8,000$ instances in `IEInstruct`. (2) **Sufficient alignment**. $\text{ADELIE}_{\text{SFT}}$ is trained based on LLAMA 2 (Touvron et al., 2023), using supervised fine-tuning (SFT) (Ouyang et al., 2022) on a mixture of `IEInstruct` and generic alignment data used in TULU 2 (Ivison et al., 2023) to maintain the model's general capabilities. We further train $\text{ADELIE}_{\text{SFT}}$ using the direct preference optimization (DPO) objective (Rafailov et al., 2023) on `IEFeedback`, a preference dataset constructed using $\text{ADELIE}_{\text{SFT}}$, resulting in $\text{ADELIE}_{\text{DPO}}$.

We comprehensively evaluate $\text{ADELIE}_{\text{SFT}}$ and $\text{ADELIE}_{\text{DPO}}$ on closed, open, and on-demand IE. Some results are shown in Figure 1. The results demonstrate that our models achieve SoTA performance compared to previous open-source models and GPT-3.5. There is no significant decline in ADELIE's general capabilities, such as MMLU (Hendrycks et al., 2021) and BBH (Suzgun et al., 2023). Moreover, we analyze several key factors of the alignment process and provide several insightful findings, such as the mixture strategy of IE and general alignment data. We hope our extensive experiments and analyses will advance research on aligning LLMs.

In summary, our contributions are threefold: (1) We construct high-quality alignment data for IE tasks: `IEInstruct` and `IEFeedback`. (2) Based on this high-quality alignment data, we develop $\text{ADELIE}_{\text{SFT}}$ and $\text{ADELIE}_{\text{DPO}}$, with advanced performance on IE tasks. (3) We conduct extensive experiments and analyses, providing meaningful insights for the research on LLM alignment.

2 Related Work

2.1 Information Extraction Tasks

Conventional IE tasks are primarily categorized into two types: closed IE and open IE. Closed IE involves extracting structured information from unstructured text, typically requiring the extracted information to conform to a predefined schema. Closed IE typically includes the following tasks: (1) Named Entity Recognition (NER), which aims to identify entities in text and categorizing them into types defined in a schema (Yadav and Bethard). (2) Relation Classification (RC), which classifies the relationship into a predefined type between two mentioned entities in the text (Han et al., 2020). (3) Relation Extraction (RE), which aims to extract entities and their relations end-to-end (Zhong and Chen). (4) Event Detection (ED), which extracts event triggers and classifies them into predefined types (Wang et al., 2020). (5) Event Argument Extraction (EAE), which aims to extract arguments, e.g., time, for events (Wang et al., 2023c). (6) Event Extraction (EE), which aims to extract events and their arguments in end-to-end paradigm (Peng et al., 2023b). (7) Event Relation Extraction (ERE), which extracts coreference, temporal, causal, and hierarchical relationships between events (Wang et al., 2022c). Open IE aims to extract n-ary relation tuples from text, without relying on a predefined schema (Zhou et al., 2022).

Beyond closed IE and open IE, Jiao et al. (2023) proposed on-demand IE, aimed at extracting user-desired information from unstructured text, such as extracting the shape and taste of fruits, and organizing it into a structured tabular format. On-demand IE is more flexible and aligns with real-world user demand. This paper covers all these IE tasks, aiming to enhance the model's ability to address these tasks through sufficient alignment.

2.2 LLMs for Information Extraction

LLMs often fall short on IE tasks (Li et al., 2023a; Han et al., 2023) due to the complex specifications of these tasks (Peng et al., 2023a). Consequently, numerous works have been proposed to

enhance LLMs' understanding of IE task specifications to improve their performance. These works are primarily divided into three aspects: (1) Prompt engineering (Pang et al., 2023; Guo et al., 2023; Wei et al., 2023b; Wang et al., 2023a; Wan et al., 2023; Zhang et al., 2023; Xie et al., 2023), aims to enhance the model's performance on IE tasks by providing sufficient prompts, such as incorporating guidelines information. Typically, these methods do not involve training model parameters. (2) Code LLMs (Guo et al., 2023; Sainz et al., 2023; Bi et al., 2023; Li et al., 2023c; Wang et al., 2023d), which adopt the Code LLMs' capabilities of understanding structured information on IE tasks, often perform better than natural language LLMs. (3) Multi-task fine-tuning (Lu et al., 2022; Wang et al., 2022a, 2023b; Sainz et al., 2023; Chen et al., 2023; Zhou et al., 2023), which trains LLMs on multiple IE datasets, enhancing the models' performance on IE tasks, especially in cross-task scenarios. These works do not sufficiently align LLMs with IE tasks, due to the lack of diverse alignment data. These trained LLMs also exhibit a decline in general capabilities. In this paper, we aim to sufficiently align LLMs on IE tasks with rich alignment data without compromising their general capabilities.

3 Alignment Data Construction

This section introduces the construction process of `IEInstruct`. The process mainly consists of 3 steps: IE data collection (§ 3.1), input construction (§ 3.2), and answer generation (§ 3.3). Details of data construction are shown in appendix A.

3.1 IE Data Collection

We first collect multiple IE datasets, including closed IE (Xu et al., 2023), open IE (Liu et al., 2022), and on-demand IE (Jiao et al., 2023), covering various domains, such as general, financial, and biomedical domains. We filter out 80% of NA data, which does not contain information needing extraction. To balance different datasets, we employ the examples-proportional mixture (Wei et al., 2022b), with a dataset size limit of $5,000$. The data collection information is shown in Figure 2.

3.2 Input Construction

We construct diverse input to better align LLMs on IE tasks. As shown in Figure 3, the input primarily consists of an instruction and a piece of input text. The instruction usually includes 3 components: task description, schema description, and output

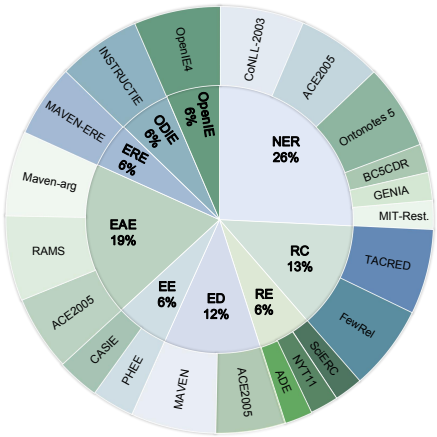

Figure 2: IE tasks, datasets, and respective proportions in `IEInstruct`.

format description. The schema description is only used in closed IE tasks, as open IE and on-demand IE do not include a schema. Some inputs also include several demonstrations, i.e., input-output exemplars, for enhancing few-shot in-context learning capabilities. We introduce the augmentation process of the 3 components of instructions and the construction of few-shot demonstrations.

Task Description For each IE task, we first manually craft 10 task descriptions. Then we adopt GPT-3.5 (OpenAI, 2022) to generate 20 more descriptions. Specifically, to enrich the diversity of generated descriptions, similar to Self-Instruct (Wang et al., 2022d), we employ an iterated generation process, which uses 3 manually written descriptions and 2 generated descriptions as the prompt for GPT-3.5 to generate a new description. Finally, we manually verify the generated descriptions and filter out those with hallucinations.

Schema Description For closed IE tasks, inspired by GoLLIE (Sainz et al., 2023), we augment the schema descriptions, i.e., category information, from 3 aspects: (1) Schema shuffling and sampling. We randomly shuffle the order of categories in the schema and select a random subset of 1 to the maximum number of categories to include in the instruction. This technique aims to prevent model overfitting on the schemata in the training corpora, forcing the model to only output categories present in the input schema. (2) Incorporation of guidelines. Guidelines are definitions of the schema, which can enhance the model's ability to understand the schema definition, thereby improving the model's zero-shot generalization capabilities on un-

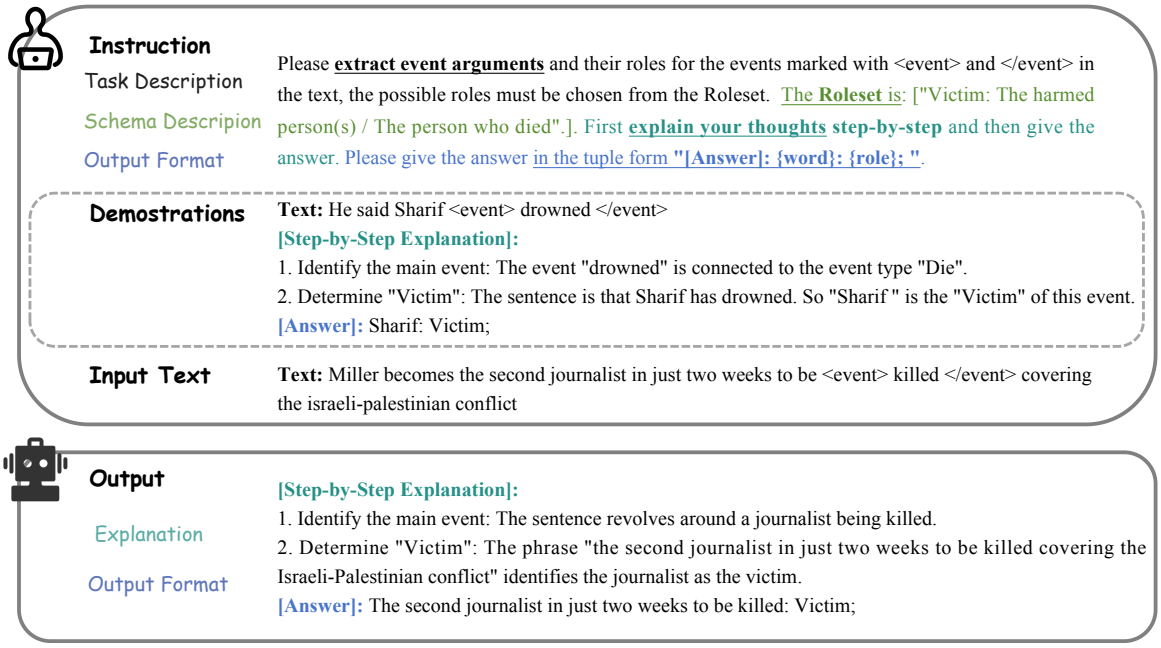

Figure 3: An example of the input and output in `IEInstruct`. 50% of the data in `IEInstruct` includes in-context demonstrations. The instruction consists of the descriptions of task, schema, and output format. The output consists of an explanation (for 30% of the instances in `IEInstruct`) and the answer adhering to the format in instruction.

seen tasks (Sainz et al., 2023). Therefore, we add guidelines information to 20% of the data in the training corpora. Similar to GoLLIE (Sainz et al., 2023), we also include several examples for each category in the guidelines. The remaining data does not include guidelines to prevent the model from memorizing schema definitions and to enhance data diversity. (3) Replacing categories with symbols. We randomly replace category names with symbols (e.g., `LABEL_1`) to prevent the model from overfitting to category names (Sainz et al., 2023) and enhance the model's in-context learning ability (Wei et al., 2023a).

Output Format Description LLMs sometimes struggle to follow the required output format in IE tasks (Han et al., 2023). To enhance the model's ability to follow format requirements, we introduce various output format descriptions in the instructions, requiring the model to output accordingly. Specifically, for each closed IE and open IE task, there are mainly 3 types of formats: (1) Triplet format, specifying output in various triple formats, e.g., *(head entity; relation; tail entity)* or *(head entity; tail entity; relation)* for relation extraction. (2) JSON format, requiring the model to output JSON formatted results. (3) Natural language format, without specific format requirements, allowing the model to output in natural language. The construction process of outputs corresponding to format requirements is detailed in § 3.3. On-demand IE does not involve output format descriptions, as its output is typically in a fixed Markdown format.

Few-shot Demonstrations Finally, to enhance the model's few-shot in-context learning capabilities, we augment the training corpus with few-shot demonstration inputs. Specifically, we randomly select 50% of the training data and add 1 to 8 randomly sampled examplars to the original input. These examplars consist of a piece of input text and the output result, with the output format adhering to the requirements in the instruction. For each instance, the demonstrations are randomly sampled and shuffled to prevent the model from overfitting to fixed demonstrations.

3.3 Answer Construction

We construct corresponding outputs according to the format requirements in the instructions generated in § 3.2. Specifically, for each closed IE and open IE task, the outputs include 3 formats: (1) Triplet format. Following Wang et al. (2022b), we convert the output into serialized triplet form. For outputs containing multiple triplets, we randomly shuffle the order of triplets to mitigate potential

order bias (Li et al., 2023b). (2) JSON format. We devise a set of JSON formats for each task and transform the answers into corresponding JSON data. (3) Natural language format. We manually write several templates for natural language outputs for each task and construct corresponding natural language outputs based on these templates. For on-demand IE, we adopt the original answers in their datasets (Jiao et al., 2023).

To enhance the model's intensive understanding of IE task procedures, we augment a subset (10%) of instances with Chain-of-Thought (CoT) (Wei et al., 2022c) explanations for closed and open IE. To generate high-quality CoT explanations, we input both the input text and its ground truth answer to GPT-4. Specifically, we first sample $1,000$ instances for each task. We then use the text input and its corresponding answer as inputs to generate CoT explanations using GPT-4. We randomly select 200 instances to assess the quality of the CoT explanations and find that GPT-4 generally generates effective and informative step-by-step thoughts for the final answers.

4 Model Training

This section introduces the alignment training process, including SFT (Ouyang et al., 2022) and DPO (Rafailov et al., 2023) training. More training details are placed in appendix B.

For the SFT training, to preserve the model's general capabilities during alignment, we utilize the general alignment corpora used by TULU 2 (Ivison et al., 2023). Specifically, we **mix** `IEInstruct` ($83,585$ instances) and $320,000$ instances of general alignment corpora as the training dataset. We adopt **LLAMA 2** (Touvron et al., 2023) as the backbone model and train the model for $6,306$ gradient steps, resulting in $\text{ADELIE}_{\text{SFT}}$.

After the SFT phase, we continue to train $\text{ADELIE}_{\text{SFT}}$ using the DPO objective. We first construct DPO training data, i.e., preference pairs (a preferred answer and a dispreferred answer). The original training objective of DPO requires online sampling of preference pairs from the model after SFT (Rafailov et al., 2023) with human annotation. In practice, some works also use human-annotated offline preference pairs for training, such as those sampled from other more powerful models (Ivison et al., 2023). In our implementation, to obtain more diverse data, we used a mix of online and offline data. Unlike previous work where preference pairs need human annotation, there exists ground truth for IE and hence the preference pairs can be automatically constructed. Therefore, similar to Chen et al. (2024), we use the model itself outputs and original ground truths without needing extra human-annotated preference pairs, which is akin to self-improvement (Huang et al., 2023) and can sufficiently minimize manual involvement and conserves labors. Specifically, we employ the BLEU (Papineni et al., 2002) score as the metric[1] to automatically construct preference pairs. We sample the output of $\text{ADELIE}_{\text{SFT}}$ 5 times for an instance with the sampling temperature as 1.0. If the difference between the highest and lowest BLEU scores exceeds 10%, we treat the corresponding outputs as a preference pair, where the higher BLEU output is the preferred answer. We denote this data as online data. We also take the lowest BLEU output as the dispreferred answer and the ground truth as the preferred answer, and denote this data as offline data. Finally, we create `IEFeedback`, containing 3k online preference pairs and 7k offline preference pairs. Then, using the DPO objective, we train for additional 937 gradient steps on $\text{ADELIE}_{\text{SFT}}$ to obtain $\text{ADELIE}_{\text{DPO}}$.

5 Experiments

5.1 Experimental Setup

Baselines For closed IE, we primarily compare 3 categories of models: (1) General open-source LLMs, including LLAMA 2 (Touvron et al., 2023), a powerful foundation model and TULU 2 (Ivison et al., 2023), an instruction tuned LLAMA 2 model. We adopt the 7B version of these models. (2) Proprietary LLMs, including GPT-3.5 (OpenAI, 2022) and GPT-4 (OpenAI, 2023). (3) Models optimized for IE tasks, including GoLLIE (Sainz et al., 2023), a code LLM fine-tuned for IE tasks, and InstructUIE (Wang et al., 2023b), an LLM trained on multiple IE tasks. For open IE, we adopt the state-of-the-art model, OpenIE6 (Kolluru et al., 2020), as the baseline. For on-demand IE, we compare with the $\text{ODIE}_{\text{Direct}}$ model (Jiao et al., 2023), which is trained on on-demand IE training set.

Evaluation Dataset For closed IE and open IE, we utilize **held-out** datasets for evaluation, i.e., the datasets not included in the alignment corpora, to better assess the models' generalization capabili-

[1]We do not use the F1 score because some predictions are unstructured and we can not directly compute their F1 scores.

	Model	$\text{FewNERD}_{\text{NER}}$	$\text{SemEval}_{\text{RC}}$	$\text{RichERE}_{\text{ED}}$	$\text{RichERE}_{\text{EAE}}$	$\text{MATRES}_{\text{ERE}}$	AVG
Zero-Shot	GoLLIE	29.7	29.2	21.0	39.2	25.9	29.0
	InstructUIE	33.5	**43.9†**	40.8	17.4	30.2	33.2
	$\text{ADELIE}_{\text{SFT}}$	32.7	21.8	24.5	45.8	47.8	34.5
	$\text{ADELIE}_{\text{DPO}}$	32.1	22.9	26.9	47.9	47.9	35.5
Few-Shot	LLAMA 2	4.4	8.2	3.0	8.9	3.8	5.7
	TULU 2	24.4	25.1	11.8	24.4	16.8	20.5
	GoLLIE	30.0	17.5	19.1	24.3	32.6	24.7
	InstructUIE	35.6	38.3†	**42.7**	17.8	10.4	29.0
	GPT-3.5*	44.1	24.0	18.8	28.7	41.0	31.3
	GPT-4*	**52.2**	39.5	23.8	41.0	**59.0**	**43.1**
	$\text{ADELIE}_{\text{SFT}}$	39.0	33.8	38.1	**54.2**	48.0	42.6
	$\text{ADELIE}_{\text{DPO}}$	37.9	34.2	39.7	53.5	48.1	42.7

Table 1: F1 scores (%) of investigated LLMs on held-out closed IE datasets. The highest scores are in **bold** and the second highest are underlined. * means the scores of the models are sourced from Peng et al. (2023a). † indicates that InstructUIE has been trained on the SemEval training set.

ties on IE tasks. Specifically, for closed IE, we employ 4 commonly used datasets: the NER dataset FewNERD (Ding et al., 2021), the RC dataset SemEval (Hendrickx et al., 2009), the ED and EAE dataset RichERE (Song et al., 2015), and the ERE dataset MATRES (Ning et al., 2018). For open IE, we use the CaRB (Bhardwaj et al., 2019) and ROBUST (Qi et al., 2023) datasets. For on-demand IE, we employ InstructIE (Jiao et al., 2023).

Evaluation Setup For closed IE and open IE, we adopt zero-shot and few-shot (4-shot for closed IE and 5-shot for open IE) in-context learning for evaluation. The few-shot demonstrations are randomly sampled from the corresponding training set. For on-demand IE, we adopt zero-shot evaluation the same as in the original paper (Jiao et al., 2023). For LLAMA 2, TULU 2, GoLLIE, and InstructUIE, we re-evaluate them using the same demonstrations. The results for GPT-3.5, GPT-4, OpenIE6, and $\text{ODIE}_{\text{Direct}}$ are obtained from previous work. Regarding evaluation metrics, we report F1 scores and employ the same calculation method as previous work. For details, please refer to Peng et al. (2023a) for closed IE, Qi et al. (2023) for open IE, and Jiao et al. (2023) for on-demand IE. More evaluation details are placed in appendix C.

5.2 Experimental Results

Results on Closed IE The results on held-out closed IE datasets are shown in Table 1. We can observe that: (1) $\text{ADELIE}_{\text{SFT}}$ performs significantly better than the original LLAMA 2 and surpasses all IE LLMs and GPT-3.5, even on par with GPT-4. This indicates that alignment on our rich data `IEInstruct` is effective. (2) DPO further enhances

	Model	CaRB	ROBUST	AVG
Zero-Shot	$\text{ADELIE}_{\text{SFT}}$	52.3	35.3	43.8
	$\text{ADELIE}_{\text{DPO}}$	53.0	36.6	44.8
Few-Shot	LLAMA 2	10.9	0.2	5.6
	TULU 2	32.5	11.0	21.8
	GPT-3.5*	51.6	27.5	39.6
	$\text{ADELIE}_{\text{SFT}}$	55.3	38.5	46.7
	$\text{ADELIE}_{\text{DPO}}$	**56.0**	**39.2**	**47.6**
Fine-Tuning	OpenIE6*	55.2	35.8	45.5

Table 2: F1 scores (%) of investigated LLMs on held-out open IE datasets. The highest scores are in **bold** and the second highest are underlined. * denotes the results are obtained from Qi et al. (2023).

performance. $\text{ADELIE}_{\text{DPO}}$ performs consistently better than $\text{ADELIE}_{\text{SFT}}$ across most datasets. This suggests that for extractive tasks with ground truth answers, further alignment using DPO can also self-improve model performance. However, the improvement of DPO is generally modest, possibly due to not using additional human-annotated preference pairs. We leave using human-annotated preference pairs for training DPO as future work. (3) Incorporating in-context demonstrations during the alignment process is necessary. Previous work only focuses on zero-shot capabilities and overlooks few-shot capabilities of LLMs, resulting in no significant improvement or even a decline when providing few-shot demonstrations, e.g., a 4.3% decline in F1 score for GoLLIE. In contrast, $\text{ADELIE}_{\text{SFT}}$'s few-shot performance is much better than its zero-shot performance, which suggests that $\text{ADELIE}_{\text{SFT}}$ possesses few-shot in-context learning capabilities for closed IE tasks. It demonstrates the effectiveness of including in-context demon-

Model	Table Header	Table Content	AVG
LLAMA 2	36.5	8.2	22.4
TULU 2	66.9	47.4	57.2
GPT-3.5*	**74.5**	51.4	63.0
GPT-4*	**74.5**	**59.1**	**66.8**
$\text{ODIE}_{\text{Direct}}$*	73.8	45.9	59.9
$\text{ADELIE}_{\text{SFT}}$	73.4	47.3	60.4
$\text{ADELIE}_{\text{DPO}}$	73.7	47.3	60.6

Table 3: F1 scores (%) of investigated LLMs on the on-demand IE task. The highest scores are in **bold** and the second highest are underlined. * means the scores of the models are sourced Jiao et al. (2023).

Model	Commonsense Reasoning	MMLU	BBH	AVG
$\text{FLAN-T5}_{\text{11B}}$	45.8	32.1	40.8	45.8
InstructUIE	42.5	30.4	13.1	42.5
LLAMA 2	55.5	45.7	35.7	52.2
`+General`	**56.9**	**49.3**	**41.7**	**54.3**
$\text{ADELIE}_{\text{SFT}}$	56.6	47.1	38.3	53.5
$\text{ADELIE}_{\text{DPO}}$	56.8	47.3	38.9	53.8

Table 4: Performance (%) on general benchmarks. "`+General`" is the model trained with only general alignment corpora for the same gradient steps as $\text{ADELIE}_{\text{SFT}}$. InstructUIE is trained based on $\text{FLAN-T5}_{\text{11B}}$.

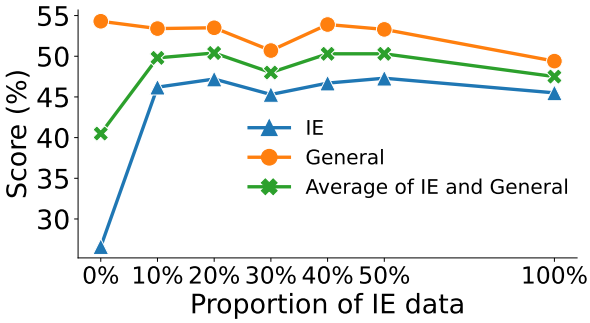

Figure 4: Scores (%) on IE tasks (average of closed IE, open IE, and on-demand IE) and general tasks (average of commonsense reasoning, MMLU, and BBH) of our model trained with varying proportions of IE data. We finally adopt a proportion of 20% to train $\text{ADELIE}_{\text{SFT}}$.

strations in the alignment process.

Results on Open IE The results on held-out open IE datasets are shown in Table 2. The observations are similar to those in closed IE. $\text{ADELIE}_{\text{SFT}}$ and $\text{ADELIE}_{\text{DPO}}$ perform much better than GPT-3.5, especially on ROBUST, a robust open IE benchmark with ubiquitous syntactic transformations (Qi et al., 2023), which demonstrates the robustness of our models on open IE. Our models even outperform the SoTA fine-tuned model, OpenIE6, demonstrating the effectiveness of alignment training.

Results on On-demand IE The results of the on-demand task are shown in Table 3. On-demand IE uses two evaluation metrics: Table header, evaluating how well the model follows instructions, and table content, assessing the extraction quality (Jiao et al., 2023). We can observe that ADELIE achieves a competitive table header score to GPT-4, which suggests that ADELIE better understands and follows user instructions. It demonstrates that the alignment process effectively aligns ADELIE with user instructions and expectations.

In general, ADELIE achieves remarkable results across all IE tasks, particularly in few-shot evaluation scenarios, which demonstrates their strong zero-shot and few-shot generalization capabilities and the effectiveness of our alignment corpora `IEInstruct` and `IEFeedback`.

6 Analysis

This section introduces further analyses of key factors in training the models (§§ 6.1 and 6.2) and analyses on few-shot ICL capabilities (§ 6.3).

6.1 Analysis on General Capabilities

Alignment may impact the model's general capabilities, namely "Alignment Tax" (Bai et al., 2022; Kim et al., 2023). We investigate the general capabilities of previous LLMs for IE and $\text{ADELIE}_{\text{SFT}}$ in this section. Specifically, we select several widely-used benchmarks for assessing general capabilities: MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2023), and Commonsense Reasoning (including HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), ARC easy and challenge (Clark et al., 2018), and OpenbookQA (Mihaylov et al., 2018)). The experimental details are placed in appendix C.3.

Table 4 presents the results. We can observe that: (1) InstructUIE suffers a significant decline in general capabilities compared to its original model, $\text{FLAN-T5}_{\text{11B}}$ (Wei et al., 2022b), which indicates that using only IE data for alignment hurts the model's general capabilities. (2) $\text{ADELIE}_{\text{SFT}}$'s performance improves compared to the original LLAMA 2. Moreover, $\text{ADELIE}_{\text{SFT}}$ performs on par with the model trained specifically on general alignment data (`+General`). This suggests that mixing general and IE alignment data can both enhance the model's general and IE capabilities and hence

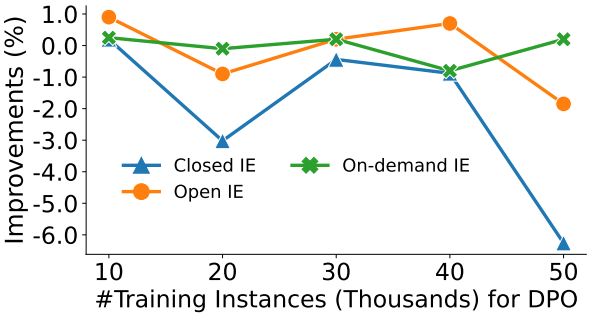

Figure 5: Performance improvements (%) of the model trained on varying scales of data, compared to ADELIE$_{\text{SFT}}$ before DPO training.

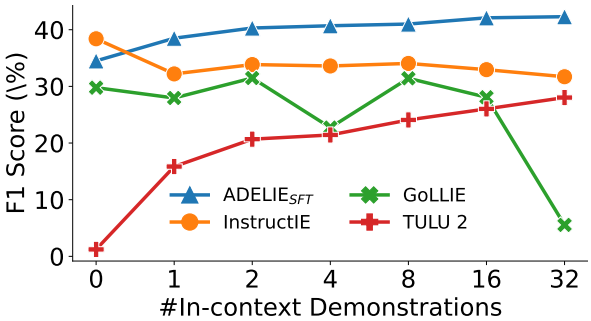

Figure 6: F1 scores (%) using a varying number of in-context demonstrations on closed IE, excluding MATRES (document-level) due to the limited context size.

mitigate the impact of "Alignment Tax". Therefore, we advocate for including `IEInstruct` in the alignment data to enhance the model's capabilities.

We further investigate the impact of data mixing strategy. Specifically, we observe the performance of models trained with varying proportions of IE data from `IEInstruct` in the overall alignment data. The results are shown in Figure 4. We can observe that: (1) There is a substantial improvement in IE tasks, even with only 10% of the training data being IE data. This suggests a lack of IE data in the existing mainstream alignment data. (2) Adding IE data in training leads to a decrease in the model's general capabilities, but this decline is limited when the proportion is below 50%. This may be due to the insufficient capacity of the 7B model, and we leave training a larger model as future work. Considering the results on both IE and general tasks, we ultimately train ADELIE on the data including 20% IE data and 80% general data.

6.2 Analysis on DPO Training

We analyze the training data construction strategy for DPO, i.e., the construction of preference pairs, each consisting of a preferred answer and a dispreferred answer. As mentioned in § 4, we adopt both offline and online data for training. The distinction lies in that both preferred and dispreferred answers of online data are sampled from ADELIE$_{\text{SFT}}$'s outputs, while the preferred answers of offline data are ground truths. We examine the impact of the proportion of offline data in the training dataset. We find that generally the model trained on 70% offline data and 30% online data performs best, with an average 47.7% F1 score across closed, open, and on-demand IE tasks. The detailed results are shown in Appendix C.4. We also explore the impact of data size on performance, as shown in Figure 5. We find that $10,000$ instances is sufficient to train the model, and using more data increases computational costs without significant improvements. This may be due to not using additional human-annotated data, leading to model overfitting. Therefore, `IEFeedback` ultimately consists of $2,996$ online and $6,989$ offline instances. We train ADELIE$_{\text{SFT}}$ on `IEFeedback` using DPO objective, resulting in ADELIE$_{\text{DPO}}$.

6.3 Analysis on Few-shot ICL Capabilities

Closed IE typically includes a schema with multiple predefined categories and hence needs more in-context demonstrations to effectively illustrate these categories (Li et al., 2024), which necessitates the few-shot in-context learning (ICL) capabilities of the model. We observe ADELIE$_{\text{SFT}}$'s few-shot ICL capabilities, as presented in Figure 6. We find that ADELIE$_{\text{SFT}}$ performs consistently better with more demonstrations, even though ADELIE$_{\text{SFT}}$ is trained with a maximum of only 8 demonstrations. In contrast, InstructUIE and GoLLIE suffer a decline with more few-shot demonstrations. This demonstrates the effectiveness of using in-context demonstrations during the alignment process.

7 Conclusion

This work introduces ADELIE, a series of LLMs aligned for information extraction tasks. ADELIE includes ADELIE$_{\text{SFT}}$, which is supervised fine-tuned on `IEInstruct` with high-quality $83,585$ instances, and ADELIE$_{\text{DPO}}$, which further trains ADELIE$_{\text{SFT}}$ on $9,985$ preference pairs (`IEFeedback`) using DPO. Extensive experiments demonstrate that ADELIE achieves impressive results on IE tasks, particularly in the few-shot setting. We hope our work can provide meaningful insights for future model alignment efforts.

Limitations

The limitations of this work are mainly threefold: (1) The preference pairs used for DPO training are automatically constructed without additional human annotation, which may limit the performance of DPO-trained models. We leave using human-annotated preference pairs for DPO training as the future work. (2) We train only with a 7B scale model due to computational limits. Employing a larger-scale model can yield better performance, but it does not impact the conclusions of this paper. (3) This paper only involves English data. In the future, we will try to support more languages, and we encourage researchers to explore aligning models for multilingual information extraction.

Ethical Considerations

We discuss potential ethical concerns of this work: (1) **Intellectual property**. Our work utilizes multiple widely-used IE datasets, and we strictly adhere to the licenses of these datasets. We will share `IEInstruct` and `IEFeedback` the CC BY-SA 4.0 license[2]. `IEInstruct` and `IEFeedback` include some data only accessible to Linguistic Data Consortium[3] (LDC) members, e.g., ACE 2005 (Christopher et al., 2005). For these parts, we will release only the data processing scripts. (2) **Intended use**. This paper introduces ADELIE, aiming to align LLMs and enhance their performance on IE tasks. (3) **Potential risk control**. `IEInstruct` and `IEFeedback` are collected and constructed based on widely-used public data and data obtained from GPT-3.5 and GPT-4. We believe that these data have been well anonymized and sanitized by their original publishers and OpenAI. We also randomly sampled 100 instances and found no sensitive data. (4) **AI assistance**. We adopt GPT-4 for paraphrasing some sentences when writing this paper.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, and et al. 2023. Gemini: A family of highly capable multimodal models. *ArXiv preprint*, abs/2312.11805.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of EMNLP*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2023. Codekgc: Code language model for generative knowledge graph construction. *ArXiv preprint*, abs/2304.09048.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of AAAI*, pages 7432–7439. AAAI Press.

Xiang Chen, Lei Li, Shuofei Qiao, Ningyu Zhang, Chuanqi Tan, Yong Jiang, Fei Huang, and Huajun Chen. 2023. One model for all domains: collaborative domain-prefix tuning for cross-domain ner. *ArXiv preprint*, abs/2301.10410.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *ArXiv preprint*, abs/2401.01335.

Walker Christopher, Strassel Stephanie, Medero Julie, and Maeda Kazuaki. 2005. ACE 2005 multilingual training corpus.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of ACL*, pages 3198–3213, Online. Association for Computational Linguistics.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, J. Guo, and Xueqi Cheng. 2023. Retrieval-augmented code generation for universal information extraction. *ArXiv preprint*, abs/2311.02962.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports.

[2] https://creativecommons.org/licenses/by-sa/4.0/

[3] https://www.ldc.upenn.edu/

Journal of Biomedical Informatics, 45(5):885 – 892. Text Mining and Natural Language Processing in Pharmacogenomics.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by ChatGPT? An analysis of performance, evaluation criteria, robustness and errors. *ArXiv preprint*, abs/2305.14450.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of APCA-IJCNL*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 94–99, Boulder, Colorado. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of ICLR*. OpenReview.net.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of EMNLP*, pages 1051–1068.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew E. Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hanna Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *ArXiv preprint*, abs/2311.10702.

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

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. 2023. Instruct and Extract: Instruction Tuning for On-Demand Information extraction. In *Proceedings of EMNLP*, pages 10030–10051.

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.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. In *Proceedings of EMNLP*, pages 13677–13700.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *Proceedings of EMNLP*, pages 3748–3761, Online. Association for Computational Linguistics.

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

Jiangnan Li, Yice Zhang, Bin Liang, Kam-Fai Wong, and Ruifeng Xu. 2023b. Set learning for generative information extraction. In *Proceedings of EMNLP*, pages 13043–13052.

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, Xuan-Jing Huang, and Xipeng Qiu. 2023c. Codeie: Large code generation models are better few-shot information extractors. In *Proceedings of ACL*, pages 15339–15353.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of NACAC*, pages 894–908, Online. Association for Computational Linguistics.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *ArXiv preprint*, abs/2404.02060.

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.

Pai Liu, Wenyang Gao, Wenjie Dong, Songfang Huang, and Yue Zhang. 2022. Open information extraction from 2007 to 2022–a survey. *ArXiv preprint*, abs/2208.08690.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *ArXiv preprint*, abs/2301.13688.

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 ACL*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of EMNLP*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of EMNLP*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of ACL*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

OpenAI. 2022. Introducing ChatGPT.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of EMNLP*, pages 15372–15389. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2023a. When does in-context learning fall short and why? A study on specification-heavy tasks. *ArXiv preprint*, abs/2311.08993.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023b. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of ACL*, pages 9206–9227.

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

Ji Qi, Chuchun Zhang, Xiaozhi Wang, Kaisheng Zeng, Jifan Yu, Jinxin Liu, Jiuding Sun, Yuxiang Chen, Lei How, Juanzi Li, et al. 2023. Preserving knowledge invariance: Rethinking robustness evaluation of open information extraction. *ArXiv preprint*, abs/2305.13981.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in NeurIPs*.

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*, abs/2310.03668.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Processing of AAAI*, pages 8732–8740. AAAI Press.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of EMNLP*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. CASIE: Extracting cybersecurity event information from text. In *Processings of AAAI*, pages 8749–8757. AAAI Press.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of EMNLP*, pages 5571–5587, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of ACL*, pages 13003–13051.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Processings of AAAI*, pages 7072–7079. AAAI Press.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

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*, abs/2307.09288.

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

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022a. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022b. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

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*, abs/2304.10428.

Xiao Wang, Wei Zhou, Can Zu, Han Xia, Tianze Chen, Yuan Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, J. Yang, Siyuan Li, and Chunsai Du. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *ArXiv preprint*, abs/2304.08085.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022c. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of EMNLP*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaozhi Wang, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, et al. 2023c. Maven-arg: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. *ArXiv preprint*, abs/2311.09105.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of EMNLP*, pages 1652–1671, Online. Association for Computational Linguistics.

Xingyao Wang, Sha Li, and Heng Ji. 2023d. Code4struct: Code generation for few-shot event structure prediction. In *Proceedings of ACL*, pages 3640–3663.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022d. Self-instruct: Aligning language model with self generated instructions. *ArXiv preprint*, abs/2212.10560.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023e. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of ACL*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *Proceedings of ICLR*. OpenReview.net.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022b. Finetuned language models are zero-shot learners. In *Proceedings of ICLR*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPs*, pages 24824–24837.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. 2023a. Symbol tuning improves in-context learning in language models. In *Proceedings of EMNLP*, pages 968–979.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023b. Zero-shot information extraction via chatting with chatgpt. *ArXiv preprint*, abs/2302.10205.

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

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *ArXiv preprint*, abs/2312.17617.

Vikas Yadav and Steven Bethard. In *Proceedings of ICCL*, Santa Fe, New Mexico, USA.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of ACL*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jiasheng Zhang, Xikai Liu, Xinyi Lai, Yan Gao, Shusen Wang, Yao Hu, and Yiqing Lin. 2023. 2iner: Instructive and in-context learning on few-shot named entity recognition. In *Findings of EMNLP*, pages 3940–3951.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. In *Proceedings of NACACL*, Online.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. 2022. A survey on neural open information extraction: Current status and future directions.

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

Appendices

A Data Collection

This section introduces details on data construction of `IEInstruct`, including details of Input Construction (appendix A.1) and Answer Construction (appendix A.2). In the data construction phase, we utilized `gpt-3.5-turbo-1106` for GPT-3.5 and `gpt-4-0125-preview` for GPT-4. The temperature parameter was set at 0.7, with all other parameters at their default settings.

A.1 Input Construction

For constructing the Task Description and Output Format Description, we initially manually wrote 10 task descriptions and 5 output format descriptions for each task. We employed GPT-4 to generate task descriptions with the same semantics but varied expressions, as well as diverse output formats.

Table 5 is an example used in the open IE task. Each generation includes five components: (1) Instruction: a description of the task. (2) Fail output: a response for when the task fails, which should correspond to the final requirement of the instruction. (3) Input template: a description of the output format in natural language, which must include multiple forms, such as triplets or natural language. (4) Output template: the output format corresponding to the input template. Table 7 details the number of augmented descriptions generated for each task.

A.2 Answer Construction

We employed GPT-4 to generate Chain-of-Thought explanations. Figure 3 illustrates examples of the explanations produced. We generate questions based on the Prompt template in Table 6. Moreover, to enhance diversity, we imposed a length constraint on the generated explanations, setting limits randomly between 70 and 200 words.

B Training Details

This section introduces the training data details (appendix B.1), and training hyper-parameters (appendix B.2). We performed each experiment once.

B.1 Datasets details

`IEInstruct` The process of constructing `IEInstruct` involves the following steps: we sampled $5,000$ instances from these raw datasets. Then, we followed the process outlined in § 3 and filtered out instances longer than $2,048$ tokens to prevent them from affecting the training effectiveness.

Finally, we compiled the `IEInstruct` dataset, which consists of a total of $83,585$ high-quality IE data instances. Table 7 shows the number of instances for each training dataset.

`IEFeedback` To generate `IEFeedback`, we sampled $50,000$ entries from raw datasets for processing in a manner similar to `IEInstruct`. The sole distinction lies in the consistency of the output format with that required by the evaluation datasets, as shown in appendix C.1, aimed at facilitating more accurate BLEU scoring. For calculating BLEU scores, we used `sentence_bleu` function from `nltk.translate.bleu_score`, with `SmoothingFunction` set to `method3`. Table 8 displays the information for the `IEFeedback` dataset, which consists of a total of $9,985$ instances.

B.2 Training Hyperparameters

SFT training To train the models, we employ supervised fine-tuning, which is the most effective method for aligning the models. The models were trained for 2 epochs with an effective batch-size of 128, a learning-rate of $2e-5$ with cosine scheduler and a warm-up phase of 0.03. To better facilitate learning in few-shot settings and document-level information extraction, the context length is set to 2048 tokens. we conduct SFT on Nvidia A100 GPUs, totaling approximately 120 GPU hours.

DPO training Similar to SFT, we train the DPO model for 3 epochs. Model is trained with a global batch size of 32. And we employ a linear learning rate scheduler with a peak learning rate of $5e-7$ and a 0.1 warm-up phase. The final $\text{ADELIE}_{\text{DPO}}$ model is initialized from the SFT model, which was trained for 2 epochs and further optimized for 3 DPO epochs. we conduct DPO on Nvidia A100 GPUs, totaling approximately 8 GPU hours.

C Experimental Details

This section introduces the details of the experiment, including the details of the evaluation(appendix C.1), the inference details of the comparison baseline (appendix C.2), and the detail results about analytical experiments (appendix C.3, appendix C.4).

Prompt

You need to follow the template list to come up with a set of diverse templates.
The task indicated by this template is the “Open Information Extraction” task.
We need to write the instruction, input format and corresponding output format template for it.
Instruction is an introduction to OpenIE tasks.
The instruction template content should include the following strings to facilitate subsequent replacement of the content: {text}.
The answer template content should include the following strings to facilitate subsequent replacement of the content: {subject}, {predicate}, {object}, {time}, {location}.
Here are the requirements:
1. Try not to repeat the verb for each template to maximize diversity.
2. The language used for the template also should be diverse. For example, use interrogative sentences, imperative sentences, etc.
3. Input and output templates ([Answer]: ..) should also be as diverse as possible.
4. Do not repeat the format of the answer template, nor repeat the examples given.
5. Input and output must correspond to each other.
6. The templates should be in English.

Template 1:
(1) Instruction: Open information extraction requires the extraction of all relations in the sentence, i.e., predicates, the subjects and objects corresponding to these relations, and the possible time and place thesis elements. Please extract all the relational tuples in the following sentence: {text}. If there are no relations in the text, please answer "NA".
(2) Fail output: NA.
(3) Input template: Please give the answer in the tuple form "[Answer]: ({predicate}; {subject}; {object}; {time}; {location})".
(4) Answer template: ({predicate}; {subject}; {object}; {time}; {location}),
...
Please follow the format given in the example to generate 1 templates.

Response

Template 6:
(1) Instruction: In the task of Open Information Extraction, your goal is to dissect the given text to unearth the underlying relationships. This involves pinpointing the subjects, predicates, and objects, as well as noting any relevant temporal or spatial details. Should the text lack relational tuples, kindly respond with ’No relationships identified.’. Text: {text}.
(2) Fail output: No relationships identified.
(3) Input template: What connections can you draw between the subject and object, including any pertinent temporal or spatial details?
(4) Answer template: Between "{subject}" and "{object}", the connection "{predicate}" is established, occurring at "{time}" and within "{location}".

Table 5: An example prompt for Open IE to generate task description and output format description.

Prompt
Please generate a step-by-step explanation for [Answer] based on [Question], and give reasons for each step.
The generated explanation should make use of the content in the [Question] as much as possible, and must be consistent with the [Answer].
It will eventually be provided at the front of the answer.
No more than {words_number} words.

[Question]: {input}
[Answer]: {output}
[Step-by-Step Explanation]:

Table 6: A prompt template for generating explanations of answers, where placeholders {words_number}, {input}, and {output} are replaced during usage.

Tasks	Datasets	Domain	#Instances	#total	#task desc.	#output format desc.
NER	CoNLL-2003 (Tjong Kim Sang, 2002)	General	5000	20000	31	15
	$ACE2005_{NER}$ (Christopher et al., 2005)	General	5000			
	Ontonotes 5 (Pradhan et al., 2013)	General	5000			
	BC5CDR (Li et al., 2016)	Biomedical	1666			
	GENIA (Kim et al., 2003)	Biomedical	1667			
	MIT-Restaurant (Liu et al., 2013)	Queries	1667			
RC	TACRED (Zhang et al., 2017)	General	5000	10000	31	15
	FewRel (Han et al., 2018)	General	5000			
RE	SciERC (Luan et al., 2018)	Scientific	3332	10000	31	15
	NYT11 (Takanobu et al., 2019)	News	3334			
	ADE (Gurulingappa et al., 2012)	Biomedical	3334			
ED	$ACE2005_{ED}$ (Christopher et al., 2005)	General	4067	9067	35	15
	MAVEN (Wang et al., 2020)	General	5000			
EE	PHEE (Sun et al., 2022)	Biomedical	2500	5000	35	15
	CASIE (Satyapanich et al., 2020)	Cybersecurity	2500			
EAE	$ACE2005_{EAE}$ (Christopher et al., 2005)	General	4420	14420	27	15
	RAMS (Li et al., 2021)	General	5000			
	Maven-arg (Wang et al., 2023c)	General	5000			
ERE	MAVEN-ERE (Wang et al., 2022c)	General	4278	4278	30	15
OpenIE	OpenIE6 (Kolluru et al., 2020)	General	5000	5000	17	15
ODIE	INSTRUCTIE (Jiao et al., 2023)	General	4904	4904	-	-

Table 7: Training Datasets for the `IEInstruct` dataset.

Tasks	Datasets	#Instances	Δ
NER	CoNLL-2003	883	0.74
	ACE2005	854	0.79
	Ontonotes 5	855	0.82
RC	TACRED	812	0.83
	FewRel	733	0.84
ED	ACE2005	753	0.83
	MAVEN	770	0.78
EAE	ACE2005	810	0.79
	RAMS	767	0.78
	Maven-arg	851	0.71
ERE	MAVEN-ERE	541	0.92
ODIE	INSTRUCTIE	617	0.87
OpenIE	OpenIE4	739	0.81

Table 8: Detailed information for the `IEFeedback` dataset. Δ represents the average difference in scores between the preferred and dispreferred answers in each dataset, with the score of the ground truth set to 1.

C.1 Evaluation Details

During the inference stage, we set the temperature to 0.01 to ensure reproducible results.

Evaluation Input Construction The input composition of the evaluation test dataset is consistent with the training set, as shown in Figure 3. The only difference is that the output format description for each task is singular to facilitate automated evaluation. Table 9 details the output format descriptions used for each task.

Evaluation Metrics In the closed IE tasks, we utilized exact matching to calculate the F1 score. For the open IE tasks on two benchmarks, we employed the same F1 calculation method as used by Qi et al. (2023). In on-demand IE tasks, following Jiao et al. (2023), we adopted a soft matching strategy for assessing table headers and used the ROUGE-L F1 score to evaluate table content.

C.2 Inference Details

We present the inference details of each baseline comparison. (1) For general open-source LLMs, including LLAMA 2 7B (`meta-llama/Llama-2-7b`[4]) and TULU 2 7B (`allenai/tulu-2-7b`[5]). The test set and the prompts used for testing are completely consistent with $\text{ADELIE}_{\text{SFT}}$. (2) For models optimized for IE tasks, including GoLLIE 7B (`HiTZ/GoLLIE-7B`[6] and InstructUIE (`ZWK/InstructUIE`[7]. We observed that these models are sensitive to prompts, and directly using the testing prompts from $\text{ADELIE}_{\text{SFT}}$ leads to a sharp decline in model performance. Therefore, while keeping the test data unchanged, we adjusted the prompts to match the official formats of these models. For GoLLIE, as it did not provide formats for ERE and RC tasks, We modified the format of the RE task for adaptation purposes.

C.3 Analysis on General Capabilities

For the MMLU task, we conducted testing using 5-shot. For the BBH task, we conducted testing using 3-shot with COT. For the remaining commonsense reasoning tasks, we employed a uniform 0-shot approach. Table 10 presents test results for detail.

C.4 Analysis on DPO Training

Table 11 presents the results in the DPO training analysis experiment. We observed a trend in which the average performance initially increased and then decreased with the increase in the offline rate. The highest performance was achieved at 0.7, reaching 47.73% (although the result displayed for 1.0 is also 47.7%, it is actually 47.68%).

[4]https://huggingface.co/meta-llama/Llama-2-7b
[5]https://huggingface.co/allenai/tulu-2-7b
[6]https://huggingface.co/HiTZ/GoLLIE-7B
[7]https://huggingface.co/ZWK/InstructUIE

[NER]
Please give the answer in the form "[Answer]: {entity}: {type}; ".
[RC]
Please give the answer in the tuple form "[Answer]: ({subject}; {relation}; {object}); ".
[ED]
Please give the answer in the form "[Answer]: {event}: {class}; ".
[EAE]
Please give the answer in the form "[Answer]: {word}: {role}; ".
[ERE]
Please give the answer in the tuple form "[Answer]: ({first event}; {relation}; {second event}); ".
[Open IE]
Please give the answer in the tuple form "[Answer]: ({predicate}; {subject}; {object}; {time}; {location})". If one or more of the last three elements does not exist, it can be omitted.

Table 9: The output format description for the hold-out tasks.

Model	MMLU	BBH	HellaSwag	ARC easy	ARC challenge	WinoGrande	OpenbookQA	SIQA	PIQA	AVG
$\text{ADELIE}_{\text{SFT}}$	47.1	38.3	57.3	78.6	46.9	69.3	32.8	32.9	78.5	53.5
$\text{ADELIE}_{\text{DPO}}$	47.3	38.8	57.5	78.9	47.3	69.2	33.0	33.1	78.8	53.8
LLAMA 2	45.7	35.7	57.1	76.3	43.3	69.1	31.6	32.9	77.9	52.2
`+General`	49.3	41.7	57.9	78.7	47.4	69.4	33.0	32.8	78.8	54.3
$\text{FLAN-T5}_{\text{11B}}$	32.1	40.8	46.4	62.4	34.7	54.7	19.2	31.8	71.3	45.8
InstructUIE	30.4	13.1	39.6	58.0	31.0	50.9	17.8	33.4	66.7	42.5
IE Proportion=0.1	46.3	41.1	57.5	76.8	45.4	70.2	32.6	33.0	78.1	53.4
IE Proportion=0.3	31.7	38.3	55.1	75.7	43.9	69.9	31.2	32.9	78.1	50.7
IE Proportion=0.4	47.3	39.0	57.7	79.4	47.8	69.1	32.8	33.4	78.3	53.9
IE Proportion=0.5	47.7	39.0	57.8	77.6	44.4	70.6	31.0	33.5	78.2	53.3
IE Proportion=1.0	38.9	23.2	56.5	74.1	40.0	69.5	31.6	32.9	77.5	49.4

Table 10: The performance of the models on general tasks in the analysis study for general capabilities.

	Models	$\text{FewNERD}_{\text{NER}}$	$\text{SemEval}_{\text{RC}}$	$\text{RichERE}_{\text{ED}}$	$\text{RichERE}_{\text{EAE}}$	$\text{MATRES}_{\text{ERE}}$	CaRB	ROBUST	Table Header	Table Content	AVG
	$\text{ADELIE}_{\text{SFT}}$	39.0	33.8	38.1	54.2	48.0	55.3	38.5	73.4	47.3	47.5
#Training	10k	37.9	34.2	39.7	53.5	48.1	56.0	39.2	73.7	47.3	47.7
	20k	34.9	36.2	33.0	46.4	47.4	54.3	37.3	73.3	47.2	45.6
	30k	36.9	34.9	38.3	53.4	47.3	55.4	38.4	73.7	47.4	47.3
	40k	36.6	34.3	38.8	52.2	46.7	55.8	39.0	72.5	46.6	46.9
Offline Data Rate	0.0	38.7	34.1	37.6	54.1	46.9	55.1	38.0	73.8	47.3	47.3
	0.3	38.4	33.2	39.8	53.8	47.3	55.2	38.2	73.7	47.5	47.5
	0.5	38.6	34.2	40.7	53.8	47.5	55.2	38.2	73.4	47.1	47.6
	0.6	38.2	33.9	38.6	53.7	47.3	55.4	38.5	73.8	47.6	47.4
	0.7	37.9	34.2	39.7	53.5	48.1	56.0	39.2	73.7	47.3	47.7
	0.8	37.9	34.4	39.2	53.8	47.8	55.6	38.8	73.6	46.9	47.6
	1.0	37.8	35.1	40.0	53.7	47.7	55.5	38.8	73.7	46.9	47.7

Table 11: The performance of models in the DPO training analysis experiment across various IE tasks. The phrase "Training Offline" denotes maintaining data proportions at 0.7 across different DPO training sets. "Offline Data Rate" refers to the proportion of offline data when the training set size is 10k.