Improving Natural Language Understanding for LLMs via Large-Scale Instruction Synthesis

Lin Yuan*, Jun Xu*, Honghao Gui*, Mengshu Sun, Zhiqiang Zhang, Lei Liang, Jun Zhou†

Ant Group, Hangzhou, China

{huiwai.yl, xujun.xj, zhongjin.ghh, mengshu.sms, lingyao.zzq, leywar.liang, jun.zhoujun}@antgroup.com

Abstract

High-quality, large-scale instructions are crucial for aligning large language models (LLMs), however, there is a severe shortage of instruction in the field of natural language understanding (NLU). Previous works on constructing NLU instructions mainly focus on information extraction (IE), neglecting tasks such as machine reading comprehension, question answering, and text classification. Furthermore, the lack of diversity in the data has led to a decreased generalization ability of trained LLMs in other NLU tasks and a noticeable decline in the fundamental model's general capabilities. To address this issue, we propose Hum, a large-scale, high-quality synthetic instruction corpus for NLU tasks, designed to enhance the NLU capabilities of LLMs. Specifically, Hum includes IE (either close IE or open IE), machine reading comprehension, text classification, and instruction generalist tasks, thereby enriching task diversity. Additionally, we introduce a human-LLMs collaborative mechanism to synthesize instructions, which enriches instruction diversity by incorporating guidelines, preference rules, and format variants. We conduct extensive experiments on 5 NLU tasks and 28 general capability evaluation datasets for LLMs. Experimental results show that Hum enhances the NLU capabilities of six LLMs by an average of 3.1%, with no significant decline observed in other general capabilities.

Introduction

NLU is a subset of natural language processing in artificial intelligence, encompassing key tasks such as machine reading comprehension, text classification, question answering, and information extraction. Recently, LLMs (Dubey et al. 2024; Yang et al. 2023) have shown impressive performance in general chat, but their language understanding ability still has shortcomings (Xu et al. 2024a; Li et al. 2023a; Cheng, Huang, and Wei 2023). Supervised instruction fine-tuning is an effective method for enhancing specific capabilities of LLMs (Sainz et al. 2023; Zeng et al. 2024). Technical reports from Llama3.1 (Dubey et al. 2024) and Qwen2 (Yang et al. 2024) emphasize that high-quality instruction data is essential for effectively aligning LLMs, and producing such data is crucial for improving model performance significantly.

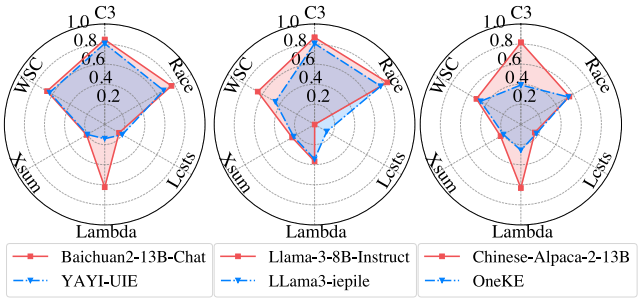

Figure 1: The existing information extraction instructions significantly reduce the performance of LLMs in NLU tasks.

Recently, there have been some works on high-quality instruction synthesis. Xu et al. (2024b) generates numerous queries through various templates, allowing current LLMs to produce responses. Cheng et al. (2024) has developed an instruction synthesis framework that converts raw pre-training text into instructional formats, substantially improving the performance of pre-trained models. Additionally, Zeng et al. (2024) has introduced an end-to-end framework that utilizes LLMs to create evolving synthesis instruction datasets. However, the instructions synthesized by these methods are domain-independent. Currently, there is a significant scarcity of instruction synthesis specifically for NLU tasks. Recently, LLMs have achieved impressive performance in unified information extraction tasks by synthesizing information extraction instructions (Xu et al. 2024a; Wang et al. 2023a; Gui et al. 2024). However, these synthesized instructions have several limitations. As illustrated in Figure 1, the language understanding performance of YAYI-UIE (Xiao et al. 2023), train on the Baichuan2-13B-Chat (Yang et al. 2023) and YAYI datasets, and Llama3-iepile (Gui et al. 2024), train on Llama-3-8B-Instruct (Dubey et al. 2024) and IEPILE, has noticeably decreased compared to their respective LLMs, with OneKE experiencing a 14.5% decline. An analysis reveals two main issues with the natural language understanding instructions developed in these cases: first, they primarily concentrate on information extraction tasks while overlooking machine reading comprehension, text classification, and question answering. Second, the instruction formats are too simplistic and fixed to make LLMs

*These authors contributed equally.

†Corresponding author

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

easily overfit these instructions. Consequently, this has significantly reduced the capabilities of LLMs trained by these instructions in the face of non-information extraction tasks or other NLU challenges.

To tackle these challenges and improve the language understanding capabilities of LLMs, we propose an effective instruction synthesis framework. First, to address the insufficient coverage of NLU tasks in prior works (Xiao et al. 2023; Gui et al. 2024; Wang et al. 2023a; Lu et al. 2022), we construct a dataset by incorporating various tasks, including information extraction, machine reading comprehension, text classification, and instruction generalist. This increased the range of task formats from 3 to 11, significantly enhancing capabilities for non-IE tasks. Second, to overcome the limitations of previous approaches that relied on a single type of instruction, we introduced innovative instruction synthesis methods, such as guidelines synthesis, preference rules synthesis, and format variants synthesis. By utilizing a diverse array of synthesis techniques, we alleviate the issue of LLMs overfitting to a single instruction, thereby helping to maintain their proficiency in non-NLU tasks even after training on the Hum dataset.

In summary, the main contribution of this work is as follows:

- We propose a framework with innovative methods for large-scale synthesis of instruction datasets. By employing guidelines synthesis, preference rules synthesis, and format variants synthesis, we address the issues of low generalization and limited instruction diversity found in NLU datasets constructed by previous works.
- We synthesize a dataset to improve the language understanding ability of LLMs and thoroughly verified that it does not significantly impact the other general capabilities of the LLMs.

Related Work

Generative Natural Language Understanding In the field of natural language understanding, mainstream tasks encompass information extraction (NER, RE, EE, OpenIE etc.), text classification (topic classification, sentiment analysis, text similarity, natural language inference, etc.), and machine reading comprehension. For information extraction, the UIE (Lu et al. 2022) framework pioneered a generation-based unified approach, effectively addressing the challenges associated with redundant models and data construction. Building on this, InstructUIE (Wang et al. 2023a) and YAYI-UIE (Xiao et al. 2023) developed a suite of information extraction instructions, implementing an instruction-based extraction framework through fine-tuning of large language models. To further enhance generalization beyond previous extraction instructions, OneKE (Gui et al. 2024) has introduced a more comprehensive and diverse set of information extraction instructions. In the realm of text classification, Wang, Pang, and Lin (2023) and Sun et al. (2023) have innovatively utilized different prompting methods to facilitate zeroshot text classification and natural language inference, respectively. For machine reading comprehension, Cheng, Huang, and Wei (2023) achieved significant performance improvements by converting extensive amounts of raw text into QA pairs before fine-tuning.

Instruction Synthesis Recent technical reports on the open-source large language models Llama 3.1 (Dubey et al. 2024) and Qwen2 (Yang et al. 2024) highlight that generating high-quality instructions is vital for training large models during both the pre-training and alignment stages. Wang et al. (2023b) proposes a framework for improving the instruction-following capabilities of pretrained language models by bootstrapping off their own generations. Li et al. (2024) exclusively utilizes a pre-curated taxonomy of human knowledge and capabilities as input and generates large-scale synthetic instruction data across all disciplines. Additionally, Dong et al. (2024) transforms the validation of instruction-following data quality into code verification, requiring LLMs to generate instructions. There are also efforts to argument instructions for IE tasks. By annotation guidelines, GoLLIE (Sainz et al. 2023) improves zero-shot information extraction, while ADELIE (Qi et al. 2024) constructs a high-quality alignment corpus for IE instructions.

Methodology

The architecture of our instruction synthesis framework is illustrated in Figure 2, which mainly consists of two parts. First, the basic instruction synthesis, which employs the structured instruction style with the field of "instruction", "schema" and "input" from existing information extraction instructions (Lu et al. 2022; Gui et al. 2024; Xiao et al. 2023; Xu et al. 2024a) and extends to other NLU tasks, such as open information extraction, machine reading comprehension, and text classification. Second, the compound instruction synthesis, which diversifies the data from the basic instruction synthesis. The main strategies for this diversification include guidelines synthesis, preference rules synthesis, and format variants synthesis.

Guidelines Synthesis

Most of the previous methods (Xu et al. 2024a; Lu et al. 2022; Wang et al. 2023a; Xiao et al. 2023) focus solely on zero-shot learning for information extraction. The instructions they design are very rigid, leading to a significant loss of in-context learning ability in large language models trained on these instructions. To address these issues, a guidelines synthesis strategy is developed. We transform the basic instructions with multiple perspectives to synthesize instructions with guidelines, which effectively prevent LLMs from overfitting and improve their language understanding capabilities. The main perspectives are as follows:

- **Description:** Add semantic explanations or typical values to a schema. For example, the schema "date" can refer to a certain date as "2024-08-15" or it can also refer to a particular month or day of the week, such as "Augest", "Thursday".
- **Example:** Providing the representative positive and negative examples in domain-specific tasks to help LLMs better follow and understand user instructions, thereby alleviating the issue of loss in-context learning capacity.

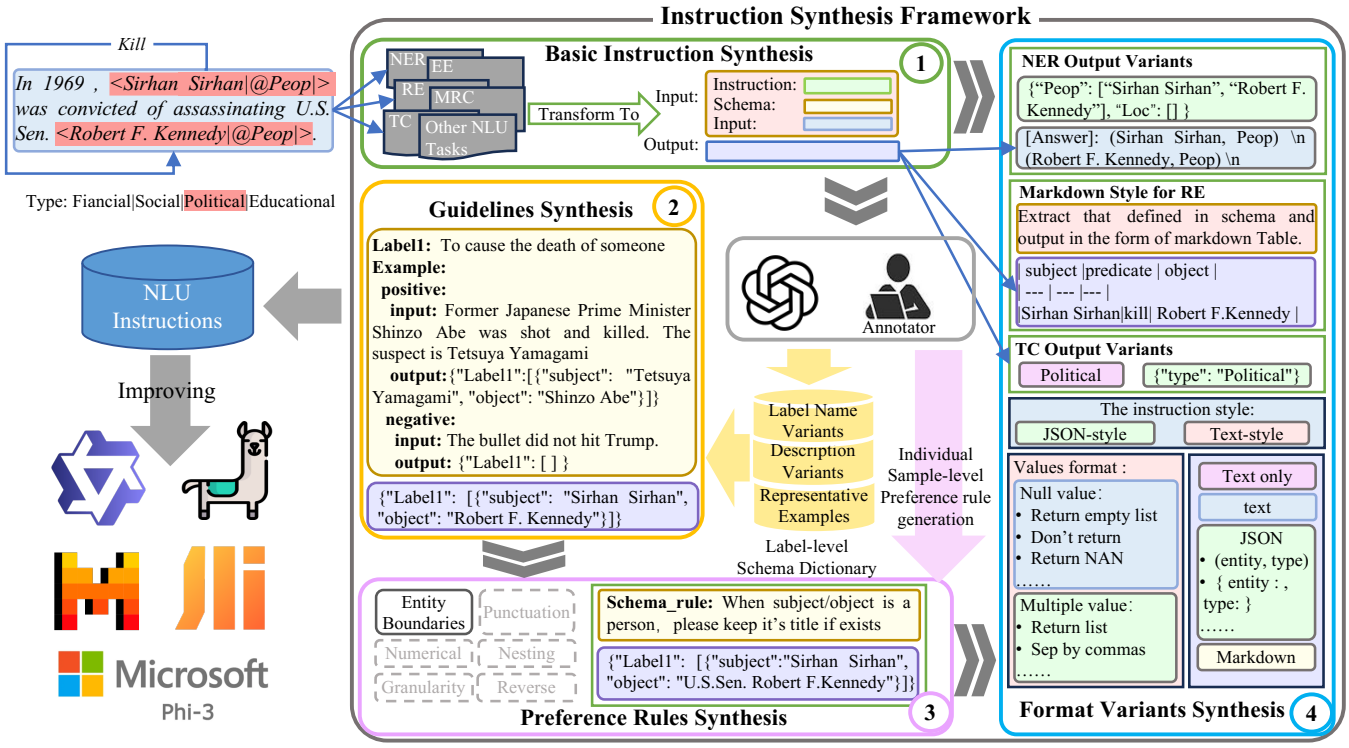

Figure 2: Overview of the natural language understanding instruction synthesis framework. The framework consists of two parts: basic instruction synthesis and compound instruction synthesis. The compound instruction synthesis mainly comprises three strategies: guidelines synthesis, preference rules synthesis, and format variants synthesis.

- **Format:** Construct various structures and formats. The structures can be hierarchical or flat. The formats include JSON, text, markdown, code, etc. By specifying the output formats of an instruction and transforming the same sample into multiple corresponding structures and formats, we further reduce the overfitting of LLMs on monotonous format of instructions.

Guideline Paraphrasing To improve the generalization ability of the instructions based on the guidelines, we make additional confusion, introduce variations, and modify the guidelines. The specific strategies are as follows:

- **Label Name Variants** : Utilizing synonyms to enhance the diversity of label name variants, for instance, the entity type "Position" may be substituted with terms such as "Title", "Job", and "Occupation".
- **Label Name Masking:** Replacing a portion of the label names within the schema with placeholders in a randomized manner encourages the model to concentrate on and comprehend the schema guidelines more effectively.
- **Description Variants**: Request LLM to generate explanations for a specific schema in a particular semantics using various expressions.
- **Representative Examples**: For each schema, we generate five positive examples, five negative examples and various representative candidates, together with other guidelines (such as descriptions and name variants) to create a comprehensive scheme dictionary. Consequently, when synthesizing an instruction, the guidelines of a certain schema, including examples, can be randomly sampled from this dictionary.

Preference Rules Synthesis

While the synthesis of guidelines can enhance the diversity of instructions, the underlying semantics of these instructions remain almost unchanged, leading to minimal variation in model outputs. To address this limitation and synthesize samples with distinct semantics, we developed a strategy named "preference rules synthesis" as depicted in Figure 3. This approach leverages existing guidelines to implement a modification strategy that utilizes GPT-4 for generating a novel labeling rule, subsequently producing entirely new outputs based on this rule. In contrast to the direct utilization of a rule library for invoking GPT-4 to create labeled samples, this methodology yields labeled samples with greater semantic diversity, effectively mitigating the risk of overfitting in LLMs and improving their capability to understand fine-grained task requirements. The proposed modification strategy is outlined as follows:

- **Entity Boundaries:** Handling of modifying prefixes and suffixes, such as *President of the United States Biden* or simply *Biden*.
- **Numerical:** Including quantities, chronological order, logical sequence, etc. For example, extracting only the

Input:
Instruction: You are an expert of NER. In order to improve the model's compliance with instructions, please give the description of the given schema, and generate the annotation rule, follow by which the original output meets. Then read the modification strategy and 2 example given as follows, determine whether it is possible to make slight adjustments to the origin rule to generate a new rule, and return annotated result that complies with the new rule.
Modification Strategy: *#Numerical (detailed content omitted)*
Examples: *#example (content omitted)*
Text: Mr. John Smith, independent director, bachelor's degree, bachelor from Harvard, Ph.D. from MIT, senior engineer with professorship.
Schema: degree
Label: ["bachelor's degree", "bachelor", "Ph.D."]

Output:
Schema Description: The name of educational qualifications and degrees.
Original Rule: Extract all educational qualifications and degrees of the individual.
New Rule: Extract the highest educational qualification of the individual. If multiple degrees exist, annotate only the highest degree.
New Label: ["Ph.D."]

Figure 3: Prompt template for preference rule annotation.

most recent position, the highest degree, or the first two companies.

- **Granularity:** Different datasets may have varying definitions for the same schema, such as organizational entities being limited to companies only.
- **Punctuation:** Addressing punctuation, book titles, numerical units, and so on. For example, *500 $* vs. *500*; *"War and Peace"* (with quotation marks) vs. *War and Peace* (without quotation marks) .
- **Nesting:** Resolving issues of nested entities. For instance, whether *Beijing* within *Beijing Sport University* should be considered a geographical entity.
- **Reverse:** The position of subject and object of a relation can be exchanged according to the semantics definition of relation label. Such as [*James Cameron*, *direct* (is the director of), *Avatar*] VS. [*Avatar*, *direct* (directed by), *James Cameron*].

Format Variants Synthesis

The previous instruction structure (Li et al. 2023b; Gui et al. 2024; Xu et al. 2024a) is constrained to a single output format, such as JSON, code, or plain text. However, numerous NLU tasks do not adhere to a singular representational style. For example, tasks such as machine reading comprehension and text classification do not align well with the JSON format, as they often struggle to define appropriate "keys" within the JSON structure. This restriction to a sole JSON instruction format poses considerable limitations on the language understanding capabilities of LLMs. To address this challenge, as show in Figure 2, we extend the output formats for identical samples to include JSON, text, markdown, and other styles. Additionally, we integrate prompts for various output styles within the input instructions, thereby transforming a singular sample into multiple representations. To mitigate the risk of LLMs becoming overfitted to a specific style, we generate multiple outputs for each output style. For instance, in the NER task, we define different candidates for producing empty results, such as "", "NAN", and []. By varying the format specification in the input instructions and selecting a diverse array of candidate outputs, we significantly enhance the variety of sample formats, ultimately alleviating the overfitting challenges faced by LLMs.

Instruction Statistics

Based on the framework mentioned above, a synthesized dataset of 2,812,832 instructions is generated. As illustrated in Figure 4, the entire dataset encompasses the following tasks: NER (23%), RE (29%), SPO (11%), EE (5%), EET (3%), EEA (2%), OpenIE (4%), KGE (12%), MRC (2%), and TC (1%), with an additional 8% IG (Instruction Generalist is included to prevent LLMs from losing its chat capability). All synthesis instructions are divided into two categories: basic instructions and compound instructions. Basic instructions account for 55% of the total. Compound instructions make up 45% and include at least one type of instruction diversity synthesis strategy (guidelines synthesis, preference rules synthesis, format variants synthesis). The total number of compound instructions is 1,261,658, in which 1,152,470 contain guidelines, 34,770 apply preference rules synthesis, and 108,091 use format variants synthesis. Due to overlaps among these strategies, the total data volume is less than the sum of the data for each individual strategy. The definitions, examples of instructions, and data source distributions for each task, along with examples of basic and compound instructions for each task, are detailed in the Appendix.

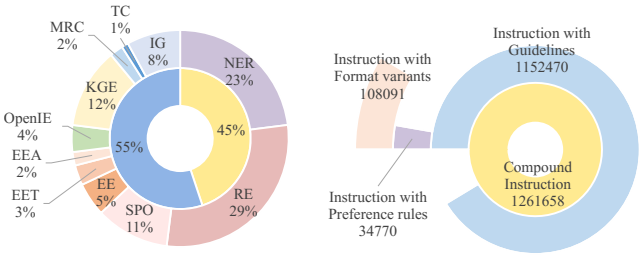

Figure 4: The source of the Hum dataset and the distribution of synthesis instructions.

Experimental Settings

Datasets

To evaluate the effectiveness of the Hum dataset for natural language understanding, we perform zero-shot experiments on five NLU datasets: CrossNER (Liu et al. 2021) for named entity recognition, FewRel (Han et al. 2018) for relation extraction, CCF Law for event extraction, C3 (Sun et al. 2020)

	CrossNER		FewRel		CCF Law		C3		IMDB		Avg	
	B	C	B	C	B	C	B	C	B	C	B	C
GPT-4	54.75	42.41	19.43	30.21	51.12/51.22	52.74/57.95	94.60	95.60	93.40	93.00	62.67	57.27
Qwen2	48.11	12.03	3.82	24.81	4.63/5.64	30.93/35.51	80.80	88.80	89.80	89.00	45.53	49.57
Llama2	29.78	0.07	0.34	4.56	0.00/0.00	12.88/11.78	12.00	39.00	39.60	78.20	16.34	26.83
Baichuan2	40.40	11.10	2.26	6.72	0.00/0.85	27.64/30.32	39.00	74.20	83.60	82.40	33.14	40.68
Llama3	0.00	0.00	0.00	0.00	0.00/0.00	10.70/13.25	67.20	76.80	90.20	90.00	31.48	35.76
OneKE	**52.22**	**59.94**	**33.93**	39.14	**62.85/62.02**	**61.86/63.88**	68.13	67.04	82.26	**90.28**	59.80	63.85
YAYI-UIE	50.39	20.75	**36.09**	16.96	12.87/59.42	38.08/40.96	**35.60**	**76.20**	49.40	**92.80**	41.53	45.85
LLama3-iepile	**51.48**	52.30	23.76	21.71	56.82/57.91	**56.73/54.54**	51.80	0.80	86.60	0.00	54.20	26.09
$\text{Hum}_{\text{Qwen2}}$	50.86	58.14	26.90	45.07	64.96/61.85	61.96/67.68	90.20	91.86	89.40	89.60	64.15	69.41
$\text{Hum}_{\text{Llama2}}$	50.68	55.86	32.92	**45.62**	66.57/52.29	64.62/58.05	**73.40**	**84.20**	**89.00**	87.97	**61.10**	**67.00**
$\text{Hum}_{\text{Baichuan2}}$	**50.57**	**56.14**	20.96	**31.95**	**65.01/64.42**	**55.19/56.76**	24.00	73.80	**91.40**	90.89	**50.33**	**61.75**
$\text{Hum}_{\text{Llama3}}$	49.42	**56.41**	**31.62**	**43.39**	**62.08/61.48**	51.63/50.85	**82.20**	**80.80**	**92.00**	**91.00**	**63.40**	**64.57**

Table 1: Zero-shot testing for tasks related to natural language understanding. The same colored background indicates that the base model is identical. Bold text indicates that the same base model performs best. The evaluation metric used is the F1 score. B denotes basic instructions in information extraction style, and the instruction format is the same as that of OneKE and Llama3-iepile. C refers to compound instructions, which may include guidelines, rules, or multiple formats. In the CCF Law dataset, x/y represent the metrics for trigger and argument, respectively.

for machine reading comprehension, and IMDB (Maas et al. 2011) for text classification. The examples of the basic and compound instructions of these five tasks are detailed in the Appendix. Additionally, to determine if the Hum dataset adversely affects LLMs, we conduct zero-shot testing across seven dimensions (language understanding, tool utilization, general knowledge, professional knowledge, coding, math, and reasoning) using a total of 28 datasets. We employ the same experimental settings as in previous work.

Comparison Methods

We categorize the comparison methods into two groups. The first group includes train-free models, such as GPT-4 (API), Qwen2 (Qwen2-7B-Instruct), Llama2 (Chinese-Alpaca-2-13B), Baichuan2 (Baichuan2-13B-Chat), Llama3 (Meta-Llama-3-8B-Instruct), Mistral (Mistral-7B-Instruct-v0.2), and Phi3 (Phi-3-medium-4k-instruct). The second group comprises supervised fine-tuned models like YAYI-UIE (based on Baichuan2), Llama3-iepile (based on Llama3), and OneKE (based on Llama2). **YAYI-UIE** builds upon InstructUIE (Wang et al. 2023a) to develop a cohesive and comprehensive framework for IE instructions. This framework is subsequently refined through supervised fine-tuning with the large language model Baichuan2-13B-Chat (Yang et al. 2023), resulting in a unified model capable of chat interactions. Meanwhile, **Llama3-iepile** aims to enhance the generalization capabilities of YAYI-UIE by integrating a broader variety of instructions. It has achieved better generalization on multiple datasets with Meta-Llama-3-8B-Instruct (Dubey et al. 2024). Additionally, **OneKE** utilizes the constructed IEPILE dataset and conducts extensive supervised fine-tuning with Chinese-Alpaca-2-13B (Cui, Yang, and Yao 2023), leading to a LLM that demonstrates improved generalization in IE tasks. All experimental results for these models are obtained through a re-evaluation based on the officially released models and using the same instructions.

Implementation Details

To mitigate the impact of different models utilizing various LLMs, we perform supervised fine-tuning with Hum across multiple LLMs. We consistently apply LoRA for this fine-tuning, with a LoRA rank and alpha both set at 64 and a dropout rate of 0.05. The batch size is established at 320, accompanied by a learning rate of 5e-5. The input length is configured to 1500 tokens, while the output length is capped at 500 tokens. We utilize an Adam optimizer with weight decay at a rate of 1e-4 for training. The learning rate warm-up proportion is set to 0.1, alongside a dropout rate of 0.1. Additionally, the temperature for adjusting next token probabilities is fixed at 0.2, with the topmost probable tokens summing to a probability of 0.95. The training is conducted using the LlamaFactory (Zheng et al. 2024) framework, leveraging 32×H100 GPUs, 384 CPU cores, and 3.2TB of memory.

Experimental Results

Overall Results

We fine-tune the Hum dataset on LLMs and conducted zero-shot evaluations across five NLU tasks. As highlighted in Table 1, our model achieves superior average performance compared to other models. In the instruction tests within the basic style, it improves by 1.3%, 8.8%, and 9.2% over OneKE, YAYI-UIE, and Llama3-iepile, respectively. For the compound instructions, the improvements are even more significant at 3.2Moreover, the supervised fine-tuning performs with Qwen2 significantly enhance the performance of the Hum dataset on these NLU tasks, showing notable gains in both basic and compound instruction scenarios. Interestingly, while OneKE showcases a commendable generaliza-

tion in IE tasks such as CrossNER, FewRel, and CCF Law, its effectiveness drops sharply in non-IE NLU tasks. This dip in performance appears to stem from OneKE's tendency to overfit the specific instructions for IE tasks, resulting in substantially lower outcomes in various other evaluations compared to Llama3-iepile and YAYI-UIE. In contrast, models fine-tune on the Hum dataset demonstrate marked advantages over train-free LLMs across NLU tasks, regardless of whether the instructions are IE-style or compound. Among these tasks, LLMs trained on the Hum dataset (Qwen2, Llama2, Llama3) consistently surpass the performance of GPT-4.

Instruction	CrossNER	FewRel
basic style	50.86	26.90
+ example	58.09	40.53
+ description	53.04	41.75
+ example & description (Hum)	58.14	45.07

Table 2: Analysis of instruction forms in in-context learning.

LLMs have excellent in-context learning capabilities. By providing specific examples and descriptions, we can effectively engage the model's cognitive abilities, leading to enhanced overall performance. As indicated in Tables 1 and 2, when instructions are presented in the basic style (B) during inference, the model achieves moderate results on CrossNER, FewRel, and various other NLU datasets. However, incorporating examples or descriptions of the content to be extracted leads to significant performance gains on OneKE, YAYI-UIE, and Hum. Furthermore, the combination of both examples and descriptions (C) enables the model to deliver even stronger results.

Model Ablation Studies

We construct four datasets: one excluding guidelines synthesis instructions, one omitting preference rule synthesis instructions, one lacking format variants synthesis instructions, and one that retained all instructions. Note that, in order to maintain the same amount of instructions as Hum, for the first three datasets, we don't simply delete the corresponding compound instructions, but instead convert them to the basic instructions. We then conduct supervised fine-tuning on these datasets using Qwen2. The results of our experiments are presented in Table 3. Notably, the removal of preference rules has the most substantial effect on the model, leading to a marked decrease in performance across all four NLU tasks. Additionally, the absence of format variants synthesis instruction cause a significant decline in the model's performance in both CrossNER and FewRel. This indicates that integrating format variants synthesis instruction can help the LLM avoid overfitting to information extraction instructions. Simultaneously, it is evident that these strategies exhibit minimal fluctuations in the CrossNER and C3 datasets. This can primarily be attributed to the relatively straightforward nature of these two tasks, which diminishes the observable impact of our instruction synthesis strategy.

Model	CrossNER	FewRel	CCF Law	C3
$\text{Hum}_{\text{Qwen2}}$	58.14	45.07	61.69/67.68	91.86
- Guidelines	60.50	41.20	59.76/64.65	89.65
- Rules	57.90	44.36	58.50/56.53	90.40
- Format	56.83	37.11	64.47/61.12	91.60

Table 3: Ablation experiments on various instruction data.

	C3	WSC	XSum	Lambda	Lcsts	Race
GPT-4	95.10	74.00	20.10	65.50	12.30	92.35
Qwen2	92.27	66.35	18.68	62.39	13.07	**88.37**
Llama2	81.70	50.96	23.29	63.26	15.99	55.64
Baichuan2	**84.44**	66.35	20.81	62.43	16.54	76.85
Llama3	**86.63**	**65.38**	25.84	36.72	0.09	**83.76**
Mistral	**67.29**	30.77	21.16	59.98	0.78	**73.46**
Phi3	68.60	**42.31**	**0.60**	**71.74**	3.47	73.18
OneKE	39.29	46.15	19.99	25.69	17.91	54.59
YAYI-UIE	80.55	63.46	19.95	14.12	20.20	67.78
Llama3-iepile	80.55	45.19	24.10	34.56	13.97	76.14
$\text{Hum}_{\text{Qwen2}}$	**92.88**	**70.19**	**31.33**	**66.16**	**18.53**	88.17
$\text{Hum}_{\text{Llama2}}$	**82.36**	**63.46**	**24.51**	**65.22**	**17.51**	**68.48**
$\text{Hum}_{\text{Baichuan2}}$	84.11	**66.35**	**21.51**	**62.64**	**17.27**	**77.18**
$\text{Hum}_{\text{Llama3}}$	83.40	62.50	**26.72**	**54.07**	**18.45**	81.16
$\text{Hum}_{\text{Mistral}}$	47.29	**39.42**	**21.54**	**69.09**	**17.14**	72.42
Hum_{Phi3}	**85.21**	25.94	0.36	71.24	**15.49**	**74.00**

Table 4: Enhancement of natural language understanding capabilities in different LLMs by Hum. The experimental results are based on the open-compass framework and tested using the "gen" mode. The evaluation metrics for C3, WSC, Lambda, and Race are ACC. XSum and Lcsts are measured using ROUGE-1. Race includes Race-middle and Race-high, and their average is taken.

Hum For Natural Language Understanding

We fine-tune six different LLMs using Hum data and evaluate them across seven dimensions.

As illustrated in Table 5, models trained on the Hum dataset, such as Llama2, Llama3, Mistral, and Phi3, show an improvement in average performance across multiple dimensions. However, there is a noticeable decline in average performance for Qwen2 and Baichuan2. When comparing against models like YAYI-UIE (based on Baichuan2), Llama3-iepile (based on Llama3), and OneKE (based on Llama2), our synthesized data substantially outperformed these in multiple dimensions. Notably, tasks related to language understanding show significant improvements across all LLMs, with an average increase of 3.1%. The models, as shown in Table 4, improve significantly on tasks such as Lcsts, Lambada, Xsum, and WSC, which are similar to information extraction tasks as they require extracting answers from the original text. In contrast, C3 and Race are multiple-choice question-answering tasks, and the Hum dataset lacks this type of data, leading to less noticeable results. For other dimensions, results are mixed with some showing improvements and others showing declines. It is noteworthy that in

	Language Understanding	Tools	General Knowledge	Professional Knowledge	Coding	Math	Reasoning	Avg
GPT-4	59.89	86.44	78.59	74.23	68.10	68.60	76.65	73.21
Qwen2	56.86	**76.03**	**71.52**	**77.73**	**62.46**	**65.03**	**67.58**	**68.17**
Llama2	48.47	**45.68**	51.72	**46.98**	**23.37**	17.85	**49.65**	40.53
Baichuan2	54.57	48.25	**60.82**	**55.90**	**25.89**	16.62	45.31	**43.91**
Llama3	49.74	56.17	62.85	55.97	**55.17**	**52.37**	**59.13**	55.91
Mistral	42.24	42.47	58.29	**48.01**	25.47	28.22	47.83	41.79
Phi3	43.32	**41.05**	55.50	52.09	**45.23**	**63.10**	44.21	49.21
OneKE	33.96	30.24	31.85	31.67	10.16	1.47	32.74	24.58
YAYI-UIE	44.34	33.89	55.56	50.58	23.70	10.02	**50.10**	38.31
Llama3-iepile	45.75	50.13	56.38	48.79	44.96	46.02	54.36	49.48
$\text{Hum}_{\text{Qwen2}}$	**61.21**	71.51	71.12	77.46	60.48	59.90	67.33	67.00
$\text{Hum}_{\text{Llama2}}$	**53.59**	45.58	**51.90**	46.68	21.56	**17.96**	49.13	**40.91**
$\text{Hum}_{\text{Baichuan2}}$	**54.84**	**49.88**	60.77	55.55	23.57	**16.66**	43.73	43.57
$\text{Hum}_{\text{Llama3}}$	**54.38**	**59.71**	**64.92**	**56.93**	51.55	50.51	58.38	**56.63**
$\text{Hum}_{\text{Mistral}}$	**44.48**	**55.49**	**60.12**	47.53	**33.76**	**30.09**	**52.28**	**46.25**
Hum_{Phi3}	**45.37**	38.60	**57.59**	**55.96**	42.42	61.98	**51.14**	**50.44**

Table 5: Performance evaluation of Hum in multiple dimensions across different LLMs. For each dimension, the average value of different datasets is taken as the reported value. The detailed dataset for each dimension can be found in the appendix.

evaluations across multiple dimensions, there is no comprehensive decline observed in any single dimension. This disparity is largely attributed to our synthesized data focusing solely on language understanding, coupled with secondary SFT on instruct/chat versions of the models, which affect the general capabilities of the base models. Future work will involve synthesizing a broader variety of data to address these limitations.

Case Study

A typical compound instruction for relation extraction is shown in Figure 5. The LLMs are asked to extract instances of the relation "located in or next to body of water", the description is given in schema to indicate the semantic range of the relation: the subject is a location and the object is the body of water. Two examples are provided (due to space limitations, the content of the examples is omitted) to describe the instances that should be extracted in practice. The output format can also be determined based on the output style of examples. The results of the same input instruction from GPT-4, Llama2, OneKE and $\text{Hum}_{\text{Llama2}}$ are listed. The Raz de Sein is a stretch of water, the La Vieille, Petite lighthouses lighthouses and le le de Sein are locations. Thus in the GPT-4 result, it has made a directional error of the subject and object. For OneKE, it may unable to understand the description and examples, thus it fails to extract and relation individuals from the text. The output of Llama2 is omitted since it is too long with the chain of thought, which also makes the result hard to be parsed. Thus we thought Llama2 is failed to understand the output format from the given examples. Finally for the result of $\text{Hum}_{\text{Llama2}}$, it extracts one valid relation instance and out put it in the required format.

Instruction: You are an expert in relationship extraction. Please extract relationship triples that match the schema definition from the input. Return an empty list for relationships that do not exist. Please respond in the format of a JSON string.You can refer to the example for extraction.
Schema: [{"relation": "located in or next to body of water","description": "Relation between location and body of water denotes geographical connectivity. Example: Port of Hull located next to River Hull ."}],
Examples: *#content omitted*
Input: The Raz de Sein is bounded by the La Vieille and Petite Vieille lighthouses and by the shoreline of the le de Sein.

Output:
GPT-4: {"located in or next to body of water": [{"subject": "Raz de Sein", "object": "shoreline of le de Sein"}]}
Llama2: The answer is too long and the content is omitted.
OneKE: {"located in or next to body of water": []}
$\text{Hum}_{\text{Llama2}}$: {"located in or next to body of water": ["subject": "La Vieille", "object": "Raz de Sein"]}

Figure 5: The performance of different large language models on the same compound instruction.

Conclusion

In this paper, we propose a novel instruction synthesis framework to create high-quality instructions aimed at enhancing the language understanding capabilities of LLMs. We find that our synthesized Hum data significantly outperforms previous methods in NLU tasks, and notably improves the language understanding abilities of LLMs while incurring minimal knowledge loss in other dimensions. Through ablation experiments, we discover that our proposed methods (guidelines synthesis, preference rules synthesis, and format variants synthesis) significantly enhance the model's generalization ability. Our instruction synthesis method is simple to implement and can be easily adapted for instruction synthesis across various tasks.

References

Cheng, D.; Gu, Y.; Huang, S.; Bi, J.; Huang, M.; and Wei, F. 2024. Instruction Pre-Training: Language Models are Supervised Multitask Learners. *CoRR*, abs/2406.14491.

Cheng, D.; Huang, S.; and Wei, F. 2023. Adapting Large Language Models via Reading Comprehension. *CoRR*, abs/2309.09530.

Cui, Y.; Yang, Z.; and Yao, X. 2023. Efficient and Effective Text Encoding for Chinese LLaMA and Alpaca. *CoRR*, abs/2304.08177.

Dong, G.; Lu, K.; Li, C.; Xia, T.; Yu, B.; Zhou, C.; and Zhou, J. 2024. Self-play with Execution Feedback: Improving Instruction-following Capabilities of Large Language Models. *CoRR*, abs/2406.13542.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; Goyal, A.; Hartshorn, A.; Yang, A.; Mitra, A.; Sravankumar, A.; Korenev, A.; Hinsvark, A.; Rao, A.; Zhang, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Gui, H.; Yuan, L.; Ye, H.; Zhang, N.; Sun, M.; Liang, L.; and Chen, H. 2024. IEPile: Unearthing Large-Scale Schema-Based Information Extraction Corpus. *CoRR*, abs/2402.14710.

Han, X.; Zhu, H.; Yu, P.; Wang, Z.; Yao, Y.; Liu, Z.; and Sun, M. 2018. FewRel: A Large-Scale Supervised Few-shot Relation Classification Dataset with State-of-the-Art Evaluation. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 4803–4809. Association for Computational Linguistics.

Li, H.; Dong, Q.; Tang, Z.; Wang, C.; Zhang, X.; Huang, H.; Huang, S.; Huang, X.; Huang, Z.; Zhang, D.; Gu, Y.; Cheng, X.; Wang, X.; Chen, S.; Dong, L.; Lu, W.; Sui, Z.; Wang, B.; Lam, W.; and Wei, F. 2024. Synthetic Data (Almost) from Scratch: Generalized Instruction Tuning for Language Models. *CoRR*, abs/2402.13064.

Li, H.; Zhang, Y.; Koto, F.; Yang, Y.; Zhao, H.; Gong, Y.; Duan, N.; and Baldwin, T. 2023a. CMMLU: Measuring massive multitask language understanding in Chinese. *CoRR*, abs/2306.09212.

Li, P.; Sun, T.; Tang, Q.; Yan, H.; Wu, Y.; Huang, X.; and Qiu, X. 2023b. CodeIE: Large Code Generation Models are Better Few-Shot Information Extractors. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 15339–15353. Association for Computational Linguistics.

Liu, Z.; Xu, Y.; Yu, T.; Dai, W.; Ji, Z.; Cahyawijaya, S.; Madotto, A.; and Fung, P. 2021. CrossNER: Evaluating Cross-Domain Named Entity Recognition. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 13452–13460. AAAI Press.

Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 2022. Unified Structure Generation for Universal Information Extraction. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 5755–5772. Association for Computational Linguistics.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In Lin, D.; Matsumoto, Y.; and Mihalcea, R., eds., *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 142–150. The Association for Computer Linguistics.

Qi, Y.; Peng, H.; Wang, X.; Xu, B.; Hou, L.; and Li, J. 2024. ADELIE: Aligning Large Language Models on Information Extraction. *CoRR*, abs/2405.05008.

Sainz, O.; García-Ferrero, I.; Agerri, R.; de Lacalle, O. L.; Rigau, G.; and Agirre, E. 2023. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. *CoRR*, abs/2310.03668.

Sun, K.; Yu, D.; Yu, D.; and Cardie, C. 2020. Investigating Prior Knowledge for Challenging Chinese Machine Reading Comprehension. *Trans. Assoc. Comput. Linguistics*, 8: 141–155.

Sun, X.; Li, X.; Li, J.; Wu, F.; Guo, S.; Zhang, T.; and Wang, G. 2023. Text Classification via Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, 8990–9005. Association for Computational Linguistics.

Wang, X.; Zhou, W.; Zu, C.; Xia, H.; Chen, T.; Zhang, Y.; Zheng, R.; Ye, J.; Zhang, Q.; Gui, T.; Kang, J.; Yang, J.; Li, S.; and Du, C. 2023a. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. *CoRR*, abs/2304.08085.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023b. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 13484–13508. Association for Computational Linguistics.

Wang, Z.; Pang, Y.; and Lin, Y. 2023. Large Language Models Are Zero-Shot Text Classifiers. *CoRR*, abs/2312.01044.

Xiao, X.; Wang, Y.; Xu, N.; Wang, Y.; Yang, H.; Wang, M.; Luo, Y.; Wang, L.; Mao, W.; and Zeng, D. 2023. YAYI-UIE: A Chat-Enhanced Instruction Tuning Framework for Universal Information Extraction. *CoRR*, abs/2312.15548.

Xu, J.; Sun, M.; Zhang, Z.; and Zhou, J. 2024a. ChatUIE: Exploring Chat-based Unified Information Extraction Using Large Language Models. In Calzolari, N.; Kan, M.; Hoste,

V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/-COLING 2024, 20-25 May, 2024, Torino, Italy*, 3146–3152. ELRA and ICCL.

Xu, Z.; Jiang, F.; Niu, L.; Deng, Y.; Poovendran, R.; Choi, Y.; and Lin, B. Y. 2024b. Magpie: Alignment Data Synthesis from Scratch by Prompting Aligned LLMs with Nothing. *CoRR*, abs/2406.08464.

Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; Yang, F.; Deng, F.; Wang, F.; Liu, F.; Ai, G.; Dong, G.; Zhao, H.; Xu, H.; Sun, H.; Zhang, H.; Liu, H.; Ji, J.; Xie, J.; Dai, J.; Fang, K.; Su, L.; Song, L.; Liu, L.; Ru, L.; Ma, L.; Wang, M.; Liu, M.; Lin, M.; Nie, N.; Guo, P.; Sun, R.; Zhang, T.; Li, T.; Li, T.; Cheng, W.; Chen, W.; Zeng, X.; Wang, X.; Chen, X.; Men, X.; Yu, X.; Pan, X.; Shen, Y.; Wang, Y.; Li, Y.; Jiang, Y.; Gao, Y.; Zhang, Y.; Zhou, Z.; and Wu, Z. 2023. Baichuan 2: Open Large-scale Language Models. *CoRR*, abs/2309.10305.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024. Qwen2 Technical Report. arXiv:2407.10671.

Zeng, W.; Xu, C.; Zhao, Y.; Lou, J.; and Chen, W. 2024. Automatic Instruction Evolving for Large Language Models. *CoRR*, abs/2406.00770.

Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z.; and Ma, Y. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. *CoRR*, abs/2403.13372.