AbsInstruct: Eliciting Abstraction Ability from LLMs through Explanation Tuning with Plausibility Estimation

Zhaowei Wang[1], **Wei Fan**[1], **Qing Zong**[1], **Hongming Zhang**[2], **Sehyun Choi**[1],
Tianqing Fang[1], **Xin Liu**[3], **Yangqiu Song**[1], **Ginny Y. Wong**[4], **& Simon See**[4]
[1]Department of Computer Science and Engineering, HKUST
[2]Tencent AI Lab, Bellevue, USA, [3]Amazon.com Inc, Palo Alto, USA
[4]NVIDIA AI Technology Center (NVAITC), NVIDIA, Santa Clara, USA
{zwanggy, yqsong}@cse.ust.hk, {gwong, ssee}@nvidia.com

Abstract

Abstraction ability is crucial in human intelligence, which can also benefit various tasks in NLP study. Existing work shows that LLMs are deficient in abstract ability, and how to improve it remains unexplored. In this work, we design the framework AbsInstruct to enhance LLMs' abstraction ability through instruction tuning. The framework builds instructions with in-depth explanations to assist LLMs in capturing the underlying rationale of abstraction. Meanwhile, we introduce a plausibility estimator to select instructions that are more consistent with the abstraction knowledge of LLMs to be aligned. Then, our framework combines abstraction instructions with general-purpose ones to build a hybrid dataset. Extensive experiments and analyses[1] demonstrate that our framework can considerably enhance LLMs' abstraction ability with strong generalization performance while maintaining their general instruction-following abilities.

1 Introduction

Abstraction ability is central to human cognition (Minsky, 1980), which is identifying shared traits among items to build a broader concept, like deriving the concept of "beverage" from "coffee" and "tea." With this ability, we can derive general rules and principles from past experiences, which enables us to adeptly navigate new situations in our daily life (Russell and Norvig, 2010; Saitta and Zucker, 2013). In NLP, building abstraction resources has long been a vital challenge to which the community has devoted many efforts (Hosseini et al., 2018; He et al., 2022).

Among them, Wang et al. (2023c) built the first comprehensive benchmark, AbsPyramid, of abstract concepts for nouns, verbs, and events. In this benchmark, models are asked to detect the validity of an abstract concept, as shown in Figure 1. Their evaluations on the benchmark reveal that abstraction remains challenging even for state-of-the-art LLMs. For example, ChatGPT (OpenAI, 2022) only modestly exceeds majority voting and substantially trails behind fine-tuned smaller models. While prior works have explored ways for general-domain LLM alignment (Sanh et al., 2022; Ouyang et al., 2022), how to elicit the abstraction knowledge of LLMs remains unexplored.

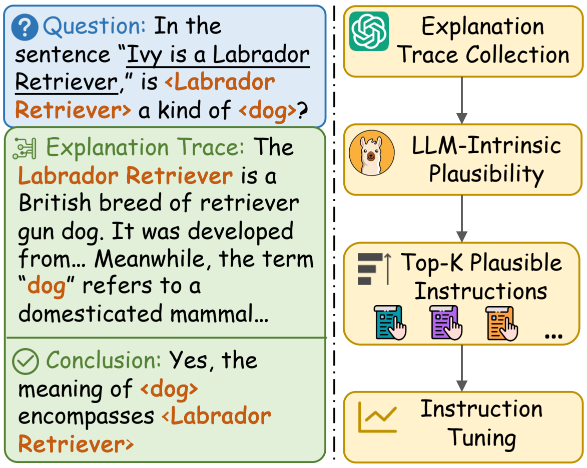
Question: In the sentence "Ivy is a Labrador Retriever," is <Labrador Retriever> a kind of <dog>?
Explanation Trace: The Labrador Retriever is a British breed of retriever gun dog. It was developed from... Meanwhile, the term "dog" refers to a domesticated mammal...
Conclusion: Yes, the meaning of <dog> encompasses <Labrador Retriever>
Explanation Trace Collection
LLM-Intrinsic Plausibility
Top-K Plausible Instructions
...
Instruction Tuning

Figure 1: An illustration of our AbsInstruct framework. We collect explanation traces for each example and design a plausibility estimator to select data that match the knowledge of an LLM to be aligned.

Nonetheless, enhancing LLMs' abstraction ability is a non-trivial task. We only observe slight improvements when gathering vanilla instructions from randomly sampled data for detecting abstract concepts. First, the responses of vanilla instructions only express the validity of abstract concepts as "Yes/No." As a result, LLMs might only grasp the surface-level styles but miss underlying rationales in deciding the validity of abstract concepts (Kung and Peng, 2023). Moreover, existing studies show that LLMs acquire most of the knowledge and abilities during pre-training (Zhou et al., 2023; Jha et al., 2023). Thus, instructions from randomly sampled data might not be consistent with

[1] The code and data are available at https://github.com/HKUST-KnowComp/AbsInstruct

the abstraction knowledge of pre-trained models for better elicitation.

To tackle those issues, we propose the framework AbsInstruct to build instructions with detailed explanation traces and well-crafted data selection, as shown in Figure 1. The framework forms explanation traces by collecting meanings of each given instance and abstract concept. These traces can help LLMs better comprehend the underlying reasoning process of detecting abstract concepts. Moreover, we introduce a plausibility estimator to select instruction data consistent with the abstraction knowledge of a pre-trained model to be aligned. The estimator assesses the plausibility score of each example based on the perplexity computed by the pre-trained model. Then, we only retain examples with higher plausibility scores, which align better with the model's knowledge. We also introduce a collection of filters based on lexical overlap, keywords, and predicted labels to ensure diversity and quality further. Ultimately, a hybrid dataset is constructed by combining instructions for abstraction detection with those in the general domain.

For evaluation, the framework first builds instructions for abstraction detection based on AbsPyramid (Wang et al., 2023c) and combines them with instructions from Alpaca (Taori et al., 2023). Next, we conduct extensive experiments and analyses of several popular LLMs instruction-tuned with our framework. The evaluation results show that applying AbsInstruct can effectively unlock LLMs' abstraction ability, with the performance surpassing existing alignment methods by a large margin of 6-10%. Also, thorough ablation studies corroborate the efficacy of explanation traces, the plausibility estimator, and various filters. Meanwhile, we conduct detailed analyses to show the robustness of our framework and the generalization ability of LLMs trained with our framework. Last but not least, the automatic and human evaluations on two general-domain instruction datasets, SuperNI (Wang et al., 2022b) and Self-Instruct (Wang et al., 2023b), manifest that our framework can enhance abstraction ability without compromising LLMs' performance of following general instructions.

2 Related Work

Abstraction has long been widely applied across various tasks, including question answering (Zheng et al., 2023a), machine translation (Padó et al., 2009), and many others (Yoshikawa et al., 2019; Khot et al., 2018; McKenna et al., 2021). While some works have studied entity abstraction (Clark et al., 2000; Wu et al., 2012; Song et al., 2015) without considering contexts, our work explores event abstraction with a few relevant fields:

Event Abstraction: This field focuses on studying abstraction within an event as context. One line of works studied extracting entailment graphs for verbs with two arguments from large corpora (Berant et al., 2011; Hosseini et al., 2018, 2019, 2021; Guillou et al., 2020; Chen et al., 2022; McKenna et al., 2021, 2023). Meanwhile, He et al. (2022) curated abstract concepts for nouns and events based on ATOMIC (Sap et al., 2019). Recently, Wang et al. (2023c) compiled a large benchmark that unifies the scopes of the abovementioned works. They collected abstraction descriptions of events and hypernyms of nouns and verbs using ChatGPT (OpenAI, 2022) and WordNet (Miller, 1995), which are then manually annotated. Their studies suggest that LLMs still struggle with abstraction knowledge with various mistakes. Thus, we present the first attempt to unlock the stronger abstraction abilities of LLMs.

Linguistic Entailment: In linguistics, the studies of event abstraction are guided by the concept of linguistic entailment (Beth, 1955; Murphy, 2010; Indarti, 2015), which is enforced by lexical semantics combined with the laws of logic. For example, *Bella is a friendly kitten* entails *Bella is a cat*, as one cannot be a friendly kitten without being a cat. Importantly, linguistic entailment contrasts with textual entailment (Dagan et al., 2005), also called NLI (Bowman et al., 2015; Conneau et al., 2018), which emphasizes what ***typicially*** can be inferred from a premise and can be fallible.

Instruction Tuning: Aligned LLMs are strongly preferred by humans over original ones (Zheng et al., 2023b; Chiang et al., 2023), and diverse methods are studied to curate instructions, such as NLP tasks (Mishra et al., 2022; Chung et al., 2022), real user requests (Conover et al., 2023), and synthetic instructions (Wang et al., 2023b). Recent studies (Mukherjee et al., 2023; Mitra et al., 2023) suggest that instructions with detailed responses can provide underlying rationales and enhance alignment efficacy. Meanwhile, several works (Zhou et al., 2023; Jha et al., 2023; Song et al., 2023) demonstrate that an LLM captures almost all the

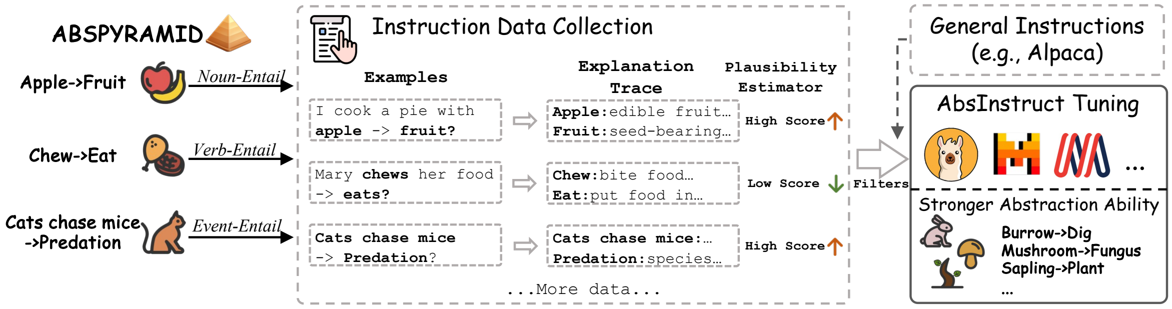

Figure 2: The overview of our AbsInstruct framework. We sample examples from AbsPyramid and collect explanation traces by prompting an LLM. Then, we design a plausibility estimator to choose examples that are more consistent with the knowledge of a model to be aligned. The framework combines abstraction instructions with general-domain ones (e.g., Alpaca) and instruction-tunes the model.

knowledge during pre-training, which can be unlocked even with a small number of instructions during alignment. Motivated by these discoveries, we collect abstraction instructions with explanation traces and try to select instructions more consistent with LLMs' knowledge for better elicitation.

3 Method

Eliciting abstraction knowledge from pre-trained LLMs can be challenging since it requires (1) underlying rationales in determining the validity of abstract concepts and (2) carefully curated instructions to better elicit the knowledge. Here, we describe the process of AbsInstruct, which builds instructions with explanation traces and employs a plausibility estimator and several filters for data selection. This pipeline is depicted in Figure 2.

3.1 Data Format Definition

Our work concentrates on detecting valid abstract concepts (Wang et al., 2023c), defined as a binary classification task. The task input is a five-element tuple in the format of (***head event***, ***entailment relation***, ***tail event***, ***instance***, ***concept***). In detail, the ***instance*** is a component of the ***head event***, which can be a noun, verb, or entire event. Then, we replace the instance with its ***concept*** to build the ***tail event***. Models are asked to decide whether the concept is a valid abstraction of the instance, where the head event linguistically entails the tail event. Here, we study three ***entailment relations*** defined on instance types: *Noun-Entail*, *Verb-Entail*, and *Event-Entail*. We provide concrete examples in Appendices D.9 and E.

The format of instruction data consists of three elements: ***instruction***, ***input***, and ***response***. An ***instruction*** outlines the task using natural language while an ***input*** and ***response*** serve as a task example. Note that the input is optional because of the blurred boundary between it and the instruction. For example, while "Give me a report about the following topic" and "global economics" can serve as separate instruction and input, we also can combine them as a sole instruction: "Give me a report about global economics."

3.2 Instruction and Input Compilation

We manually build instructions for all entailment relations: *Noun-Entail*, *Verb-Entail*, and *Event-Entail*. Since our framework introduces detailed responses with explanation traces, the instructions for each relation comprise two steps, asking LLMs to (1) consider the meanings of the given instances and concepts and (2) predict the label based on the explanation in the first step.

Next, we collect the input of abstraction detection for each relation. Our framework samples five-element tuples with balanced labels from the training set of AbsPyramid (Wang et al., 2023c). To build the input, we verbalize each tuple using prompts that ask whether the concept is a valid abstraction of the instance, given the head event as context. We provide concrete prompts for building the instructions and input in Appendix A.1.

3.3 Response Collection with Explanation

In conformity with instructions, our framework collects responses consisting of two steps: (1) the ***explanation step***, which contains the meanings of given words, and (2) the ***conclusion step***, which confirms the concept validity by comparing word meanings. The easy-to-build component is the ***conclusion step***. For each example, we verbalize the binary label as "Yes" or "No" and append a short comparison, such as "Yes, the meaning of **[cpt]** encompasses **[ins]**," where **[ins]** and **[cpt]** are two placeholders for the given instance and concept.

For the ***rationale step***, we first conduct a pilot study about using taxonomies to build explanation traces, such as WordNet (Miller, 1995), which can provide meanings of nouns and verbs. Our findings disclose two problems with using a taxonomy. First, the coverage of nouns in WordNet is inadequate. Only 6.32% of nominal phrases can be found in WordNet. For example, while "cat" is incorporated in WordNet, many specific types are absent, such as fluffy cat and ginger cat. Moreover, we need word sense disambiguation (Pradhan et al., 2007) to choose correct word meanings, which may accumulate errors in our framework. For example, the expert annotation shows that only 61.0% of WSD results from GlossBERT (Huang et al., 2019) are correct (Details in Appendix B).

To overcome those challenges, we build explanation traces with the help of an LLM. In detail, we prompt GPT4 under the zero-shot setting with the instruction asking the meaning of a given word. We collect the meanings of the instance and concept separately and then concatenate them to build the whole explanation trace. After collecting both steps, the framework constructs the whole response with the format:

```
Step1: <ins mean> Meanwhile, <cpt mean>
Step2: Yes/No, the meaning of ...
```

where **`<ins mean>`** and **`<cpt mean>`** stand for the meanings of the instance and concept. The whole response interprets and compares the given instance and concept, assisting LLMs in seizing the underlying reasoning processes. We provide concrete prompts for using GPT4 in Appendix A.2.

3.4 Example Postprocessing

After gathering many examples, we employ several filters and a plausibility estimator to select instructions. First, two quality filters are introduced to remove basic errors: the prediction filter and the keyword filter. Then, we introduce a diversity filter based on ROUGE-L (Lin, 2004) to remove similar examples. Lastly, we design a plausibility estimator to select abstraction examples consistent with pre-trained LLMs' knowledge.

Prediction Filter: A faithful explanation trace should assist LLMs in reaching the correct prediction. Therefore, given the explanation trace we built, we prompt GPT4 to predict a label for each example. Then, we discard all examples that GPT4 cannot give the right answer:

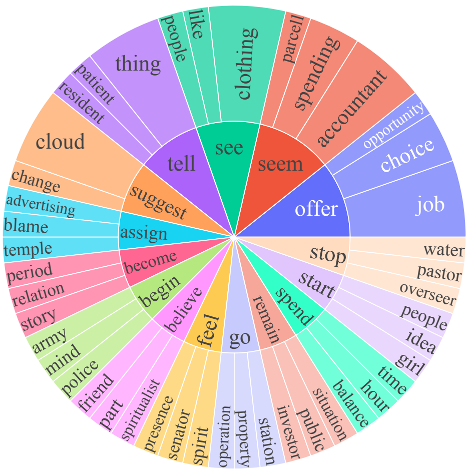

Figure 3: The 15 most common verbs (inner circle) and their top 3 direct nominal objects (outer circle) in head events of collected examples.

$$\begin{cases} \hat{y} = \theta_{LLM}(i, x, e) \\ f_{pred}(\hat{y}, y) = \mathbb{1}\{\hat{y} = y\}, \end{cases} \tag{1}$$

where θ_{LLM} signifies the parameters of GPT4 that outputs a predicted label $\hat{y}$ given the instruction i, input x, and explanation trace e. Then, the filter f_{pred} compares $\hat{y}$ with ground truth y.

Keyword Filter: We observe that GPT4 may explain the meaning of another word in the head event rather than the given one due to hallucination (See cases in Appendix E). Thus, we design the keyword filter to discard examples whose explanation trace omits its instance or concept. Take Figure 1 as an example. The explanation must contain both the keywords "Labrador Retriever" and "dog."

Diversity Filter Our framework collects a large pool of examples from ABSPYRAMID, which could result in multiple examples with similar instances or concepts. To promote diversity, a new example is added only if its ROUGE-L similarity with any existing example is below 0.7, following prior works (Wang et al., 2023b; Taori et al., 2023).

Plausibility Estimator Existing studies (Zhou et al., 2023; Jha et al., 2023) show that a model obtains its knowledge almost entirely during pretraining, which can be elicited with a modest set of examples during alignment. For better elicitation, we select examples that are more consistent with the knowledge of the pre-trained LLM to be aligned. Here, we measure the LLM-intrinsic plausibilities of each example, which is determined by

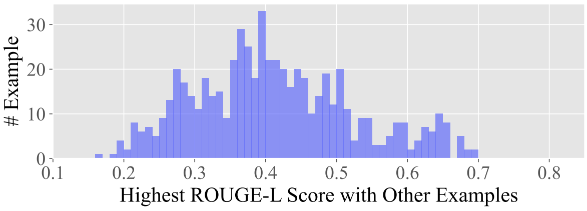

Figure 4: Distribution of the ROUGE-L scores between collected examples. For each example, we compute the highest similarity with other examples we gathered.

the model's knowledge. Concretely, the plausibility is computed as the conditional probability of the response r given the instruction i and input x:

$$Plausibility(i, x, r) = P_\theta(r|i, x), \qquad (2)$$

where θ are the parameters of the pre-trained model. Then, the framework only retains examples with top-K plausibilities. Note that we compute plausibilities based on a model's intrinsic knowledge, in contrast to those definitions on real-world knowledge (Wu et al., 2012; Chalier et al., 2020).

3.5 Mixed Alignment Data

Our framework combines the abstraction instructions we collected and general-domain instructions to build the final dataset. The dataset is then used to finetune the same model that we use to compute plausibility scores. We concatenate an instruction and the input as a prompt (See details in Appendix C.1) and train the model to generate the response in a standard supervised way.

4 Abstraction Instruction Overview

In this section, we apply AbsInstruct for inducing instruction data as a case study, with Llama2 (7B) (Touvron et al., 2023) used to estimate plausibilities. Our framework constructs 200 examples for each relation, derived from AbsPyramid.

4.1 Diversity

We identify the verb-noun structure in the head events of examples to examine the diversity of collected examples. We use the Berkeley Neural Parser (Kitaev and Klein, 2018; Kitaev et al., 2019) to parse each event and then extract the verb that is closest to the root as well as its first nominal object. 391 out of 600 head events contain such structure as other events usually are more complex, such as "PersonX began renting the space to businesses." We plot the 15 most common verbs and their top 3 direct nominal objects in Figure 3, which makes up 9.67% of the entire set. Overall, we see diverse topics and textual formats in these examples.

Quality Review Question	Yes %
Is the explanation of the instance correct?	94.7%
Is the explanation of the concept correct?	96.0%
All fields are valid	92.0%

Table 1: Data quality annotation for explanation traces generated by GPT4.

We further study the diversity of collected examples. For each example we collect, we compute its highest ROUGE-L similarity with other ones. We plot the distribution of these ROUGE-L scores in Figure 4. The results indicate a decent number of unique examples, which do not overlap much with the remaining.

4.2 Quality

To investigate the quality, we sample 150 examples and ask three experts to label the correctness of the meanings of instances and concepts (See details in Appendix B). Results in Table 1 demonstrate that most of the collected explanation traces are meaningful. While some traces may contain noise, we found that explanation traces can provide useful guidance for tuning LLMs for abstraction ability.

5 Experiment

We conduct extensive experiments and compare our framework with various baselines.

5.1 Dataset and Evaluation Metric

We study LLMs' abstraction ability on AbsPyramid, a large-scale dataset of abstraction knowledge with statistics in Appendix D.1. Our framework and baselines build examples based on five-element tuples from its training set. Meanwhile, the general-purpose instruction dataset we use is Alpaca (Taori et al., 2023), which contains 52K instructions generated with the Self-Instruct framework (Wang et al., 2023b). We mix instructions for abstraction with those general-purpose ones to fine-tune LLMs in the following experiments.

We calculate Accuracy and Macro F1-score for metrics between predicted and ground truth labels to evaluate all models' abstraction ability.

5.2 Baseline Methods

We compare our framework to three baselines and provide implementation details in Appendix C,

Methods	Backbone	Noun		Verb		Event		All	
		Acc	Ma-F1	Acc	Ma-F1	Acc	Ma-F1	Acc	Ma-F1
Random	-	50.00	49.56	50.00	49.95	50.00	48.98	50.00	49.50
LLM API (Zero)	GPT 4	79.70	77.34	57.50	54.24	69.70	63.32	68.97	64.97
	GPT 3.5	67.00	62.45	56.30	55.90	65.60	58.23	62.97	58.86
	ChatGPT	74.00	72.27	56.30	55.71	68.20	63.22	66.17	63.73
	ChatGPT (SC)	74.40	72.75	55.50	54.70	68.90	63.49	66.27	63.65
LLM API (10-shot)	GPT 4	70.50	70.49	57.30	56.88	67.20	62.91	65.00	63.43
	GPT 3.5	73.10	71.74	57.20	57.07	66.90	63.79	65.73	64.20
	ChatGPT	76.10	74.60	58.60	58.51	68.90	60.51	67.87	64.54
	ChatGPT (SC)	76.60	75.07	59.10	59.04	68.80	59.56	68.17	64.55
Alpaca (10-shot)	MPT (7B)	43.42	34.71	48.72	37.94	65.33	43.72	52.49	38.79
	Falcon (7B)	60.68	55.07	56.35	56.15	63.92	45.17	60.32	52.13
	Mistral (7B)	76.08	74.10	59.20	58.66	67.66	60.69	67.65	64.49
	Llama2 (7B)	61.96	61.94	55.53	53.19	69.71	60.24	62.40	58.46
	Llama2 (13B)	75.28	72.31	58.97	58.92	66.93	61.73	67.06	64.32
Direct Injection	MPT (7B)	63.87	63.23	53.71	52.37	51.85	51.70	56.47	55.77
	Falcon (7B)	63.48	58.54	55.27	55.16	51.21	51.14	56.66	54.95
	Mistral (7B)	74.90	74.62	59.39	59.11	59.95	59.27	64.74	64.33
	Llama2 (7B)	67.24	66.34	56.66	55.72	55.11	55.11	59.67	59.05
	Llama2 (13B)	75.04	74.09	60.04	59.91	59.26	58.44	64.78	64.15
AbsInstruct	MPT (7B)	71.34	70.89	58.63	58.63	67.52	65.16	65.83	64.89
	Falcon (7B)	66.92	66.45	57.06	56.11	69.03	64.15	64.33	62.24
	Mistral (7B)	80.59	79.85	**60.80**	**60.74**	70.96	66.54	70.78	69.04
	Llama2 (7B)	77.07	75.81	59.44	59.07	**72.72**	**68.00**	69.74	67.63
	Llama2 (13B)	**81.13**	**80.35**	60.58	60.58	71.92	67.24	**71.21**	**69.39**

Table 2: Performance of AbsInstruct and baselines on the test set of AbsPyramid. We abbreviate Accuracy and Macro F1-score to **Acc** and **Ma-F1**, respectively. We bold the best score and underline the second-best score. See Appendix D.3 for the performance on the validation set.

including learning rates, example numbers, API specifics, prompts for baselines, etc.

API-based LLM: We evaluate a series of closed-source LLMs under the zero-shot and few-shot (10-shot) settings, covering GPT3.5 (Ouyang et al., 2022), ChatGPT (OpenAI, 2022), and GPT4 (Achiam et al., 2023). In addition, we test ChatGPT with the self-consistency decoding strategy (Wang et al., 2022a).

Alpaca LLM: An intuitive method is to align open-source LLMs and test their abstraction ability with in-context learning. Here, we choose to tune LLMs with Alpaca (Taori et al., 2023), including models of MPT (7B) (Team, 2023), Falcon (7B) (Penedo et al., 2023), Mistral (7B) (Jiang et al., 2023), Llama2 (7B, 13B) (Touvron et al., 2023). For inference, we test models with ten exemplars randomly sampled from AbsPyramid.

Direct Injection: This baseline randomly samples tuples from AbsPyramid and builds examples with the vanilla prompts (in Appendix C.2), where responses are solely "Yes" or "No." Then, we mix abstraction examples with Alpaca for alignment. Similarly, the LLMs we tested are MPT (7B), Falcon (7B), Mistral (7B), and Llama2 (7B, 13B).

Models	Noun	Verb	Event	All	Δ_{All}
Llama2 (7B)	**75.81**	**59.07**	**68.00**	**67.63**	-
◇ w P-Random	69.56	58.48	66.04	64.69	↓2.94
◇ w P-Input	69.92	58.43	66.34	64.90	↓2.73
◇ w/o Q Filter	65.06	56.90	62.70	61.55	↓6.08
◇ w/o P&Q Filter	65.79	57.27	54.52	59.19	↓8.44
◇ w/o E Trace	69.98	58.25	66.27	64.84	↓2.79
◇ w/o All Parts	66.34	55.72	55.11	59.05	↓8.58

Table 3: Ablation study for AbsInstruct. Macro F1-scores are exhibited, and Δ_{All} indicates score changes. See Appendix D.4 for results of all models.

6 Main Evaluation

We present the results of each entailment relation and the average on the test set of AbsPyramid in Table 2. In general, our framework AbsInstruct can unlock stronger abstraction ability from LLMs, exceeding the performance of all baselines by a large margin. For example, Mistral (7B) tuned with our framework correctly classifies 70.78% of test examples, increasing by 6.04% compared to the

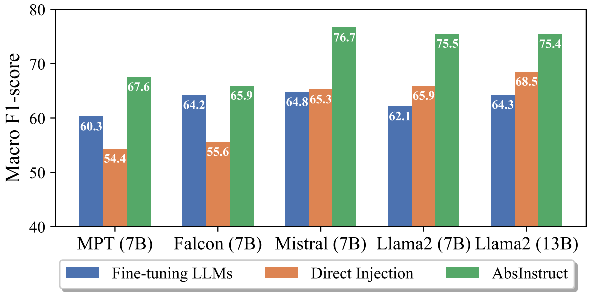

Figure 5: The out-of-domain performance on the abstractATOMIC dataset. We provide results across all metrics in Appendix D.6.

Models	Acc	Ma-F1	Δ_{Acc}	$\Delta_{\text{Ma-F1}}$
Fine-tuned on AbsPyramid				
Mistral (7B)	79.32	72.66	-	-
Llama2 (7B)	78.69	71.07	-	-
Llama2 (13B)	82.11	71.25	-	-
Direct Injection				
Mistral (7B)	85.34	74.55	↑6.02	↑1.89
Llama2 (7B)	84.29	74.00	↑5.60	↑2.93
Llama2 (13B)	85.51	76.27	↑3.40	↑5.02
AbsInstruct				
Mistral (7B)	86.61	77.80	↑**7.29**	↑5.14
Llama2 (7B)	84.31	78.76	↑5.62	↑7.69
Llama2 (13B)	**87.11**	**79.89**	↑5.00	↑**8.64**

Table 4: The performance on the Levy/Holt dataset. Δ_{Acc} and $\Delta_{\text{Ma-F1}}$ mean improvements compared to LLMs fine-tuned on AbsPyramid. We show results of all LLMs in Appendix D.6

“Direct Injection” baseline. Meanwhile, Llama2 (13B), tuned with our framework, outperforms all the API-based LLMs, even GPT4.

Our results unequivocally demonstrate that the “Direct Injection” baseline possesses limited efficacy in eliciting abstraction knowledge. For example, Falcon (7B) only achieves performance slightly higher than a random guess. Similarly, we observe that LLMs tuned with Alpaca only capture limited generalization ability in abstraction detection, even with ten exemplars. For instance, Falcon (7B) only achieves a Macro F1-score of 52.13%, lagging behind our framework by about 10 points.

6.1 Ablation Study

To better understand how to unlock abstraction ability, we conduct several ablation experiments to show the effectiveness of explanation traces, quality filters, and plausibility estimators. The results of ablation studies are presented in Table 3.

Plausibility Estimator: We conduct two experiments to verify the efficacy of the plausibility estimator. First, we remove the estimator and randomly select examples (◇ w P-Random). From the results in Table 3, we can find noticeable performance declines, verifying the plausibility estimator’s efficacy. Moreover, we consider another way to measure plausibilities instead of conditional probabilities of explanation traces. Here, we compute the probabilities of example input (◇ w P-Input). As the performance consistently drops, we can see that explanation traces play a pivotal role in selecting plausible examples.

Quality Filters: We also conduct ablation studies for quality filters, including the prediction and keyword filters. Results (◇ w/o Q Filter) show that LLMs’ performance deteriorates drastically after we remove these filters. Then, we further remove the plausibility estimator besides quality filters (◇ w/o P&Q Filter). The results, like the decline of 8.44% on average, again show the efficacy of our filters and the plausibility estimator. Meanwhile, we analyze the role of the diversity filter in Appendix D.5, where we find that our framework can collect highly diverse examples and explanation traces, even without the diversity filter.

Explanation Traces: First, we remove explanation traces and employ the vanilla prompt, also used by the “Direct Injection” baseline. The results (◇ w/o E Trace) show that LLMs cannot perform well. Further, we remove all the filters, estimator, and explanation traces (◇ w/o All Parts), where we observe greater decreases in performance. Here, Llama2 (7B) significantly drops by 8.58% in the Macro F1-score. These findings demonstrate the utility of the explanation traces we collect.

6.2 Out-of-Domain Evaluation

This section studies if our framework can generalize to other tasks requiring abstraction knowledge. We conduct experiments on two out-of-domain datasets: AbstractATOMIC (He et al., 2022) and Levy/Holt dataset (Levy and Dagan, 2016; Holt, 2018), with statistics in Appendix D.2.

AbstractATOMIC: First, we test our framework on the AbstractATOMIC dataset and treat “Direct Injection” as a baseline. We also fine-tune LLMs on AbsPyramid to test their transferring ability on AbstractATOMIC. As depicted in Figure 5, our framework can equip LLMs with broader general-

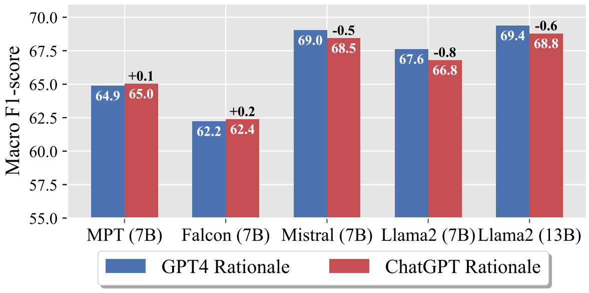

Figure 6: Macro F1-scores of our framework with ChatGPT as the source of explanation traces. We also provide performance changes. See full results across all metrics in Appendix D.7.

ization abilities. Particularly, Mistral (7B) attains a Macro F1-score of 76.7%, which is substantially higher than "Direct Injection." Also, Llama2 (7B) exhibits improvements of over 10 points compared to its fine-tuned counterpart, which demonstrates our work's essence of eliciting abstraction ability instead of fitting a specific dataset.

Levy/Holt Dataset: This dataset is primarily used to evaluate verb entailment graphs. We test the performance of models tuned with our framework and take the same baselines as AbstractATOMIC. As shown in Table 4, our framework performs better on the Levy/Holt dataset than the "Direct Injection" baseline. More generally, instruction-tuning methods can obtain better generalization than fine-tuning on ABSPYRAMID, given that instruction-tuning only needs a tiny fraction of training data. With our framework, the Macro F1-score of Llama2 (13B) improves considerably by 8.64% compared to the fine-tuned one.

6.3 Discussion of Explanation Trace

In previous sections, we collect explanation traces by prompting GPT4. Here, we evaluate our framework with explanation traces from a less advanced model, ChatGPT, to gain a deeper insight into the robustness. We plot and compare the Macro F1-scores in Figure 6. The outcomes suggest that our framework maintains its strong performance with some fluctuations below 1 point. In particular, the score of Falcon (7B) improves by only 0.2% while Llama2 (13B) declines by only 0.6%.

6.4 Task Instruction Following

Prior experiments demonstrate the effectiveness of our framework in abstraction ability. Additionally, we also evaluate the ability of LLMs to follow general-purpose instructions for NLP tasks. Here, we choose the test set of SuperNI (Wang et al., 2022b), consisting of 119 tasks with 100 examples in each task. Following previous works (Wang et al., 2023b; Xu et al., 2023), we evaluate LLMs by calculating ROUGE-L (Lin, 2004), BLEU-1/2 (Papineni et al., 2002), and Meteor (Banerjee and Lavie, 2005). For baselines, we train LLMs on the instruction dataset Alpaca (Taori et al., 2023), as it is also used in our framework. The results in Table 5 show that LLMs tuned with our framework can attain comparable scores to those fine-tuned on Alpaca. For instance, MPT (7B) obtains a slightly higher Rouge-L while Llama2 (7B) drops by only 0.09%. These findings manifest that injecting a few instructions for abstraction knowledge does not sacrifice the general ability of instruction following.

Models	R-L	B-1	B-2	Meteor	$\Delta_{\textbf{R-L}}$
ALPACA					
MPT (7B)	41.20	26.44	14.37	26.20	-
Falcon (7B)	39.38	24.21	12.88	25.19	-
Mistral (7B)	50.47	**44.66**	**26.83**	31.06	-
Llama2 (7B)	43.70	28.21	15.19	27.37	-
Llama2 (13B)	48.30	30.70	17.32	30.39	-
ABSINSTRUCT					
MPT (7B)	43.43	26.40	14.40	27.71	$\uparrow$**1.51**
Falcon (7B)	39.76	26.52	14.38	25.66	$\uparrow$0.47
Mistral (7B)	**51.22**	42.58	24.99	**31.88**	$\uparrow$0.82
Llama2 (7B)	43.35	26.80	14.29	27.28	$\downarrow$0.09
Llama2 (13B)	49.19	31.71	17.66	31.25	$\uparrow$0.86

Table 5: Performance on the test set of SuperNI. **R-L** and **B-1/2** denote ROUGE-L and BLEU-1/2. $\Delta_{\textbf{R-L}}$ means the performance changes compared to Alpaca.

Meanwhile, previous works (Ouyang et al., 2022; Zhao et al., 2023) suggest a disparity between NLP tasks and human requests. Thus, we also conduct a human evaluation on expert-curated instructions (Wang et al., 2023b) to better understand the alignment with human values. The evaluation setups and results are shown in Appendix D.8, which again manifests that our framework can preserve LLMs' general capabilities.

7 Conclusion

In this paper, we propose ABSINSTRUCT, which is the first attempt to elicit stronger abstraction abilities from pre-trained LLMs. Our framework builds instructions for abstraction detection with explanation traces and a plausibility estimator. Then, these abstraction instructions are combined with general-domain ones from Alpaca. We provide extensive experiments to demonstrate the effectiveness of our framework. Besides eliciting abstraction knowledge, future works can study how to equip LLMs with more knowledge during pre-training.

Limitations

Prior research (Zhou et al., 2023) indicates that LLMs primarily acquire their knowledge during the pre-training phase while the alignment phase only teaches LLMs about the specific subdistribution of interactions with users. In this work, we mainly focus on the alignment phase, while it remains unclear what abstraction knowledge is captured by LLMs during pre-training. Following previous works of knowledge probing (Hou et al., 2023; Sun et al., 2023), future research can probe recent LLMs, like Llama2, to better understand this question and explore how to equip LLMs with more abstraction knowledge during pre-training.

Meanwhile, instruction tuning only elicits the existing knowledge of pre-trained LLMs. We leave for future works about equipping LLMs with new abstraction knowledge through other techniques, like knowledge editing (Wang et al., 2023a; Zhang et al., 2024; Hase et al., 2023), retrieval augmented generation (Lewis et al., 2020; Gao et al., 2023) and memory mechanism (Madaan et al., 2022).

Ethics Statement

We evaluate the abstraction ability on ABSPYRAMID (Wang et al., 2023c), which is a free and open-source dataset. The out-of-domain (OOD) datasets, namely AbstractATOMIC (He et al., 2022) and Levy/Holt (Levy and Dagan, 2016; Holt, 2018), are also freely available and open-source. Meanwhile, all the instruction datasets are released under the Apache-2.0 License, including Alpaca (Taori et al., 2023), SuperNI (Wang et al., 2022b), and SELF-INSTRUCT (Wang et al., 2023b).

Human evaluations are performed by three expert annotators with at least one year of expertise in NLP to ensure the quality. The annotation works are compensated at the hourly rate of 7.6 USD, higher than the local minimum wage.

Acknowledgements

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20), and the GRF (16211520 and 16205322) from RGC of Hong Kong. This paper was also supported by the Tencent Rhino-bird Focused Research Program. We also thank the support from NVIDIA AI Technology Center (NVAITC) and the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619.

Evert Willem Beth. 1955. Semantic entailment and formal derivability.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Yohan Chalier, Simon Razniewski, and Gerhard Weikum. 2020. Joint reasoning for multi-faceted commonsense knowledge. In *Automated Knowledge Base Construction*.

Zhibin Chen, Yansong Feng, and Dongyan Zhao. 2022. Entailment graph learning with textual entailment and soft transitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5899–5910.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/.

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

Peter Clark, John A. Thompson, Heather Holmback, and Lisbeth Duncan. 2000. Exploiting a thesaurus-based semantic net for knowledge-based search. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, pages 988–995. AAAI Press / The MIT Press.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, et al. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Liane Guillou, Sander Bijl de Vroe, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2020. Incorporating temporal information in entailment graph mining. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 60–71.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *arXiv preprint arXiv:2301.04213*.

Mutian He, Tianqing Fang, Weiqi Wang, and Yangqiu Song. 2022. Acquiring and modelling abstract commonsense knowledge via conceptualization. *arXiv preprint arXiv:2206.01532*.

Xavier Ricketts Holt. 2018. *Probabilistic Models of Relational Implication*. Ph.D. thesis, Macquarie University.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R Holt, Shay B Cohen, Mark Johnson, and Mark Steedman. 2018. Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Mohammad Javad Hosseini, Shay B Cohen, Mark Johnson, and Mark Steedman. 2019. Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746.

Mohammad Javad Hosseini, Shay B Cohen, Mark Johnson, and Mark Steedman. 2021. Open-domain contextual link prediction and its complementarity with entailment graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2790–2802.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4919.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514.

Gatri Asti Putri Indarti. 2015. Distinguishing entailment and presupposition under negation test. *LLT Journal: A Journal on Language and Language Teaching*, 18(1):27–38.

Aditi Jha, Sam Havens, Jeremey Dohmann, Alex Trott, and Jacob Portes. 2023. Limit: Less is more for instruction tuning across evaluation paradigms. *arXiv preprint arXiv:2311.13133*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. *arXiv preprint arXiv:2305.11383*.

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Multivalent entailment graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10758–10768.

Nick McKenna, Tianyi Li, Mark Johnson, and Mark Steedman. 2023. Smoothing entailment graphs with language models. In *IJCNLP-AACL*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Marvin Minsky. 1980. K-lines: A theory of memory. *Cognitive science*, 4(2):117–133.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

M Lynne Murphy. 2010. *Lexical meaning*. Cambridge University Press.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt.

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.

Sebastian Padó, Daniel Cer, Michel Galley, Dan Jurafsky, and Christopher D Manning. 2009. Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*, 23:181–193.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 87–92.

Stuart J Russell and Peter Norvig. 2010. *Artificial intelligence a modern approach*.

Lorenza Saitta and Jean-daniel Zucker. 2013. *Abstraction in Artificial Intelligence and Complex Systems*. Springer New York, NY.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.

Chiyu Song, Zhanchao Zhou, Jianhao Yan, Yuejiao Fei, Zhenzhong Lan, and Yue Zhang. 2023. Dynamics of instruction tuning: Each ability of large language models has its own growth pace. *arXiv preprint arXiv:2310.19651*.

Yangqiu Song, Shusen Wang, and Haixun Wang. 2015. Open domain short text conceptualization: a generative+ descriptive modeling approach. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3820–3826.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: How knowledgeable are large language models (llm)? aka will llms replace knowledge graphs? *arXiv preprint arXiv:2308.10168*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. https://www.databricks.com/blog/mpt-7b.

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

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, and Mengru Wang. 2023a. Zekun xi, siyuan cheng, kangwei liu, guozhou zheng, et al. easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022b. Supernaturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Zhaowei Wang, Haochen Shi, Weiqi Wang, Tianqing Fang, Hongming Zhang, Sehyun Choi, Xin Liu, and Yangqiu Song. 2023c. Abspyramid: Benchmarking the abstraction ability of language models with a unified entailment graph. *arXiv preprint arXiv:2311.09174*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, pages 481–492.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Masashi Yoshikawa, Koji Mineshima, Hiroshi Noji, and Daisuke Bekki. 2019. Combining axiom injection and knowledge base completion for efficient natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7410–7417.

Hongming Zhang, Xin Liu, Haojie Pan, Haowen Ke, Jiefu Ou, Tianqing Fang, and Yangqiu Song. 2022. Aser: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities. *Artificial Intelligence*, 309:103740.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. Aser: A large-scale eventuality knowledge graph. In *Proceedings of the web conference 2020*, pages 201–211.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2023a. Take a step back: evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

A AbsInstruct Prompts

This appendix lists the concrete prompts we use in our framework. First, we provide the prompts of building instructions, input, and responses. Then, we show the concrete prompts we use to collect word meanings from GPT4.

A.1 Prompts for Instructions and Examples

We manually collect the prompt templates of instructions, input, and responses used in our framework, shown in Table 6. Models are given five-element tuples on the AbsPyramid dataset: (***head event***, ***entailment relation***, ***tail event***, ***instance***, ***concept***).

In our prompt templates in Table 6, there are three placeholders **[head]**, **[cpt]**, and **[ins]** for head events, concepts, and instances. Specifically, **[ins]** is the same as **[head]** for *Event-Entail*. Meanwhile, we indicate the entailment relations implicitly by using different instructions for different relations. For example, for *Noun-Entail*, the instruction contains "Identify the hypernym of a specific **noun**." Note that the tail event can be built by replacing the instance with the concept in the head event. In conclusion, our prompt does not lose any information provided by five-element tuples.

A.2 Prompts for Word Meanings

To build explanation traces, we also prompt GPT4 to collect the meanings of instances and concepts in a zero-shot manner. Here, we ask GPT4 to provide meanings of given words and then detect whether the given concept is valid. The prompt is shown in Table 7. We collect the meanings of instances and concepts in the first and second steps separately. Then, we concatenate them to build explanation traces.

B Human Annotation

We conduct a few human evaluations in our study, including the Accuracy of GlossBERT (Huang et al., 2019), the quality of examples collected by our framework, and the ability of our framework to follow human instructions. In this appendix, we discuss the details of annotation and agreement between annotators.

All annotation tasks are performed by three postgraduate NLP researchers with at least one year of expertise in NLP. They understand our annotation tasks clearly and can serve as experts. Two annotators are authors of the paper, and the third

***Noun-Entail* Instruction**: Hypernyms are words with a broad meaning, which more specific words fall under. Identify the hypernym of a specific noun through the following two steps: Step 1: Let's think about meanings of those words. Step 2: Provide a "Yes" or "No" response.
***Verb-Entail* Instruction**: Hypernyms are words with a broad meaning, which more specific words fall under. Identify the hypernym of a specific verb through the following two steps: Step 1: Let's think about meanings of those words. Step 2: Provide a "Yes" or "No" response.
***Event-Entail* Instruction**: Identify abstract descriptions of specific sentences through the following two steps: Step 1: Let's think about meanings of the sentence and the abstract description. Step 2: Provide a "Yes" or "No" response.

(a) Instructions used by our framework.

***Noun-Entail* Input**: In the sentence **[head]**, does the meaning of **[cpt]** encompass **[ins]**?
***Verb-Entail* Input**: In the sentence **[head]**, does the meaning of **[cpt]** encompass **[ins]**?
***Event-Entail* Input**: Can we consider **[cpt]** as an abstract description of the sentence **[head]**?

(b) Input templates used by our framework.

***Noun-Entail*, *Verb-Entail*, and *Event-Entail* Response** **Positive Label:** Step1: **<ins mean>**. Meanwhile, **<cpt mean>**. Step2: Yes, the meaning of **[cpt]** encompasses **[ins]**. **Negative Label:** Step1: **<ins mean>**. Meanwhile, **<cpt mean>**. Step2: No, the meaning of **[cpt]** does not encompass **[ins]**.

(c) Response templates used by our framework.

Table 6: The concrete prompts we used in our AbsInstruct framework. We show the instruction, input, and response templates in each table segment. Placeholders **[head]**, **[cpt]**, and **[ins]** will be replaced with real head events, concepts, and instances. Also, **<ins mean>** and **<cpt mean>** will be replaced with the meanings of real instances and concepts.

is another NLP researcher within the same institution for a more objective perspective. The authors' involvement in the annotation process is part of their academic responsibilities, and no additional compensation is provided. The third annotator is compensated at the hourly rate of 7.6 USD, higher than the local minimum wage.

GlossBERT Accuracy: We sample 500 examples from AbsPyramid and run GlossBERT to disambiguate the given noun or verb. Three experts are asked to evaluate whether the disambiguation results are right, yielding 1500 ratings in total. The IAA score is 78.8% calculated using pairwise agreement proportion, and the Fleiss's κ (Fleiss, 1971) is 0.57.

Noun-Entail: Identify the hypernym of a specific noun. Hypernyms are words with a broad meaning, which more specific words fall under. In the sentence **[head]**, does the meaning of the new word **[cpt]** encompass the original word **[ins]**? Step 1: Let's think about the meaning of the original word. Step 2: Let's think about the meaning of the new word. Step 3: Provide a "Yes" or "No" response without other words.
Verb-Entail: Identify the hypernym of a specific verb. Hypernyms are words with a broad meaning, which more specific words fall under. In the sentence **[head]**, does the meaning of the new word **[cpt]** encompass the original word **[ins]**? Step 1: Let's think about the meaning of the original word. Step 2: Let's think about the meaning of the new word. Step 3: Provide a "Yes" or "No" response without other words.
Event-Entail: Identify abstract descriptions of specific sentences. Can we consider **[cpt]** as an abstract description of the sentence **[head]**? Step 1: Let's think about the meaning of the sentence. Step 2: Let's think about the meaning of the abstract description. Step 3: Provide a "Yes" or "No" response without other words.

Table 7: The zero-shot prompts we used for collecting meanings of instances and concepts. Placeholders **[head]**, **[cpt]**, and **[ins]** will be replaced with real head events, concepts, and instances. We collect the meanings of instances and concepts in the first and second steps separately. Then, we concatenate them to build explanation traces.

Quality of Collected Examples: We sampled 150 explanation traces collected by our framework AbsInstruct. Similarly, three experts are asked to label two aspects: the correctness of explanations for given instances and concepts. This leads to 900 total ratings (150 examples × 2 aspects × 3 annotators). The Fleiss's κ (Fleiss, 1971) is 0.62.

Human Instruction Following: There are 252 instructions in the test set of Self-Instruct (Wang et al., 2023b), which are curated manually by experts. Here, the expert annotators are asked to annotate which response is preferred between our framework and the Alapca baseline. This leads to 756 ratings for each model. The IAA score is 80.95% calculated using pairwise agreement proportion, and the Fleiss's κ (Fleiss, 1971) is 0.71.

C Implementation Details

We access open-source language models using Transformers (Wolf et al., 2020) and fine-tune them on 8 NVIDIA A100 (80G) GPUs. We fine-tune 7B

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: **{instruction}** ### Input: **{input}** ### Response:

(a) Template for examples with a non-empty input field.

Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: **{instruction}** ### Response:

(b) Template for examples with an empty input field.

Table 8: The prompt templates we used to concatenate instructions and example input. We show two templates since the input is optional. Placeholders **{instruction}** and **{input}** will be replaced with real instructions and example input.

and 13B LLMs with LoRA (Hu et al., 2021) and load them with BF16. For LoRA, we only add new parameters to attention layers with the rank and α equal to 512 and 1024. The best checkpoint is selected according to the sum of all metrics on the validation set. The batch size and training epoch are 128 and 3, respectively. We grid search learning rates of 5e-6, 1e-5, 2e-5, 3e-5 and 5e-5.

We collect rationales for about 2,000 examples for each entailment relation and keep 200 examples with the highest LLM-intrinsic plausibility after filtering. For a fair comparison, the "Direct Injection" baseline also incorporates 200 examples for each entailment relation. We discuss choices of example numbers and show that 200 is proper in Appendix C.4. For API-based LLMs, We access ChatGPT, GPT4, and GPT3.5 via OpenAI API[2]. The specific versions are `gpt-3.5-turbo-0613`, `gpt-4-1106-preview`, and `gpt-3.5-turbo-instruct-0914`. They are evaluated on one thousand examples that we randomly sampled from the test set of each entailment relation due to the trade-off between API expenses and our evaluation's precision. For self-consistency, we sample 5 responses independently for each example and take the majority vote.

Noun-Entail **Instruction**: Identify the hypernym of a specific noun and provide a "Yes" or "No" response. Hypernyms are words with a broad meaning, which more specific words fall under.
Verb-Entail **Instruction**: Identify the hypernym of a specific verb and provide a "Yes" or "No" response. Hypernyms are words with a broad meaning, which more specific words fall under.
Event-Entail **Instruction**: Identify abstract descriptions of specific sentences, and provide a "Yes" or "No" response.

(a) Instructions of the vanilla prompt.

Noun-Entail **Input**: In the sentence **[head]**, does the meaning of **[cpt]** encompass **[ins]**?
Verb-Entail **Input**: In the sentence **[head]**, does the meaning of **[cpt]** encompass **[ins]**?
Event-Entail **Input**: Can we consider **[cpt]** as an abstract description of the sentence **[head]**?

(b) Input templates of the vanilla prompt.

***Noun-Entail*, *Verb-Entail*, and *Event-Entail* Response** **Positive Label:** Yes. **Negative Label:** No.

(c) Responses of the vanilla prompt.

Table 9: The vanilla prompt we used in the "Direct Injection" baseline. We show the instruction, input, and response templates in each table segment. Placeholders **[head]**, **[cpt]**, and **[ins]** will be replaced with real head events, concepts, and instances.

Noun-Entail: Identify the hypernym of a specific noun and provide a "Yes" or "No" response. Hypernyms are words with a broad meaning, which more specific words fall under. In the sentence **[head]**, does the meaning of **[cpt]** encompass **[ins]**?
Verb-Entail: Identify the hypernym of a specific verb and provide a "Yes" or "No" response. Hypernyms are words with a broad meaning, which more specific words fall under. In the sentence **[head]**, does the meaning of **[cpt]** encompass **[ins]**?
Event-Entail: Identify abstract descriptions of specific sentences, and provide a "Yes" or "No" response. Can we consider **[cpt]** as an abstract description of the sentence **[head]**?

Table 10: The zero-shot prompt we used in the "API-based LLM" baseline. Placeholders **[head]**, **[ins]**, and **[cpt]** will be replaced with real head events, instances, and concepts.

C.1 Prompts for Concatenation

We should concatenate the instructions and input as a prompt for our framework and the instruction-tuned baselines: "Alpaca LLM" and "Direct Injection." In our experiments, we employ the same prompt template as used by Alpaca (Taori et al., 2023), which is shown in Table 8.

[2] https://platform.openai.com/docs/api-reference

***Noun-Entail*:**
Instruction: You need to decide whether a hypernym of a specific noun is valid or not. Hypernyms are words with a broad meaning, which more specific words fall under.
Exemplars and test example: 1. In the sentence **[head]**$^{(1)}$, is **[cpt]**$^{(1)}$ a hypernym of **[ins]**$^{(1)}$? Yes. (No.) 2. In the sentence **[head]**$^{(2)}$, is **[cpt]**$^{(2)}$ a hypernym of **[ins]**$^{(2)}$? Yes. (No.) . . . 11. In the sentence **[head]**$^{(11)}$, is **[cpt]**$^{(11)}$ a hypernym of **[ins]**$^{(11)}$?
Verb-Entail
Instruction: You need to decide whether a hypernym of a specific verb is valid or not. Hypernyms are words with a broad meaning, which more specific words fall under.
1. In the sentence **[head]**$^{(1)}$, is **[cpt]**$^{(1)}$ a hypernym of **[ins]**$^{(1)}$? Yes. (No.) 2. In the sentence **[head]**$^{(2)}$, is **[cpt]**$^{(2)}$ a hypernym of **[ins]**$^{(2)}$? Yes. (No.) . . . 11. In the sentence **[head]**$^{(11)}$, is **[cpt]**$^{(11)}$ a hypernym of **[ins]**$^{(11)}$?
Event-Entail
Instructions: You need to decide whether an abstract description of a specific sentence is valid or not.
1. Can we consider **[cpt]**$^{(1)}$ as an abstract description of the sentence **[head]**$^{(1)}$? Yes. (No.) 2. Can we consider **[cpt]**$^{(2)}$ as an abstract description of the sentence **[head]**$^{(2)}$? Yes. (No.) . . . 11. Can we consider **[cpt]**$^{(11)}$ as an abstract description of the sentence **[head]**$^{(11)}$?

Table 11: The in-context learning prompt (10-shot) we used in the "API-based LLM" baseline. Placeholders **[head]**, **[ins]**, and **[cpt]** will be replaced with real head events, instances, and concepts.

C.2 The Vanilla Prompt of the "Alpaca LLM" and "Direct Injection" Baselines

This appendix provides the vanilla prompt used by the "Alpaca LLM" and "Direct Injection" baseline to build instructions and examples for abstraction detection, as demonstrated in Table 9. In contrast to our framework, the responses of this vanilla prompt are simply "Yes" or "No," verbalized directly from the binary labels.

C.3 The Prompt of the "API-based LLM" Baseline

We employ the same prompt as those utilized in ABSPYRAMID (Wang et al., 2023c), which exhibit considerable robustness when benchmarked against other prompts featured in the study of ABSPYRAMID. We provide zero-shot prompts used by the "API-based LLM" baseline in Table 10. The in-context learning prompts used by the "API-based LLM" baseline are shown in Table 11.

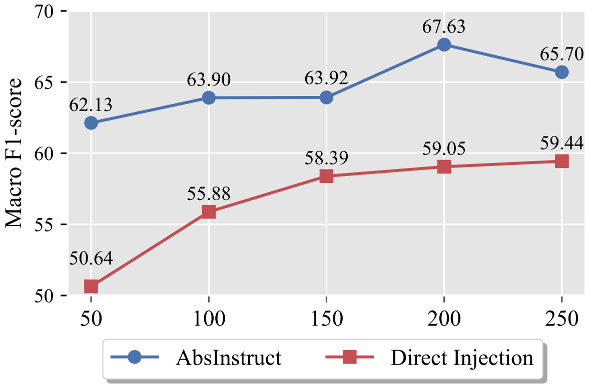

Figure 7: Macro F1-scores of different example numbers. We instruction-tune Llama2 (7B) with our ABSINSTRUCT framework and the "Direct Injection" baseline.

C.4 Discussion of Example Number K

In our experiments, we collect $K = 200$ examples of abstraction detection for each relation in our framework and the "Direct Injection" baseline. In this appendix, we study the proper values of this hyperparameter and grid search different example numbers K of 50, 100, 150, 200, and 250. Here, we combine different numbers of abstraction instructions with Alpaca and instruction-tune Llama2 (7B) with our ABSINSTRUCT framework and the "Direct Injection" baseline. We plot the performance in Figure 7. For the "Direct Injection" baseline, the Macro F1-score shows rapid improvements as the number of examples increases from 50 to 200. Then, the improvement stagnates (i.e., lower than 0.5 points) at the number of 250. Similarly, for our framework, we observe the first decrease when the example number is 250. For both methods, we can find that the improvements are small when the example numbers are higher than 200. Thus, we recommend choosing 200 examples for each relation in our framework and the "Direct Injection" baseline when considering the tradeoff between abstraction ability and general-domain ability.

D Supplementary Experiments

This appendix provides more supplementary experiments and analysis of the ABSINSTRUCT framework.

Split	Noun	Verb	Event	All	Pos %
Train	79,034	47,669	49,988	176,691	58.86
Valid	9,874	5,939	6,237	22,050	58.22
Test	9,875	5,934	6,247	22,056	59.02
Ours	200	200	200	600	50.00

Table 12: Statistics of AbsPyramid and abstraction examples collected by AbsInstruct. **Pos %** denotes positive rates of each split. Our framework samples 200 examples for each relation with balanced labels.

Datasets	# Total	# Train	# Valid	# Test
AbsAtomic	92235	75814	8027	8394
Levy/Holt	18407	N/A	5486	12921

Table 13: Statistics of AbstractATOMIC and Levy/Holt datasets. **# Total** is the number of all examples.

D.1 Abstraction Data Statistics

We study LLMs' abstraction ability on AbsPyramid (Wang et al., 2023c), a large-scale benchmark of abstraction knowledge comprising more than 221K examples. The dataset samples head events from ASER (Zhang et al., 2020, 2022) and collects abstract concepts of three components of head events: nouns, verbs, and entire events. AbsPyramid collects candidates of abstract concepts using WordNet (Miller, 1995) and ChatGPT (OpenAI, 2022), which is then manually verified. Our framework only builds 200 examples for each entailment relation based on five-element tuples from the training split. We present comprehensive statistics of AbsPyramid and our examples in 12.

D.2 Out-of-Domain Datasets Statistics

As we conduct experiments on two out-of-domain datasets: AbstractATOMIC (He et al., 2022) and Levy/Holt dataset (Levy and Dagan, 2016; Holt, 2018), we provide comprehensive statistics of these two datasets in Table 13. AbstractATOMIC samples base events from ATOMIC (Sap et al., 2019) in the commonsense domain and collects thousands of abstract concepts for nouns and entire events. Meanwhile, the Levy/Holt dataset is primarily used in the study of verb entailment graphs, where events are simplified as a verb with two entity types as arguments (i.e., *subject* and *object*).

D.3 Validation Results on Abstraction Detection

We collect the performance of our framework AbsInstruct and baselines on the validation set of the AbsPyramid in Table 23.

Models	Noun	Verb	Event	All	Δ_{All}
MPT (7B)	**70.89**	**58.63**	**65.16**	**64.89**	-
◇ w P-Random	69.57	57.89	55.26	60.90	↓3.99
◇ w P-Input	70.06	58.32	59.52	62.63	↓2.26
◇ w/o Q Filter	60.24	53.02	54.40	55.89	↓9.00
◇ w/o P&Q Filter	58.61	48.06	32.42	46.37	↓18.52
◇ w/o E Trace	65.46	54.77	63.10	61.11	↓3.78
◇ w/o All Parts	63.23	52.37	51.70	55.77	↓9.12
Falcon (7B)	**66.45**	**56.11**	**64.15**	**62.24**	-
◇ w P-Random	61.85	55.53	62.30	59.89	↓2.35
◇ w P-Input	61.25	53.92	58.95	58.04	↓4.20
◇ w/o Q Filter	52.81	39.83	58.25	50.30	↓11.94
◇ w/o P&Q Filter	59.50	50.41	59.36	56.42	↓5.82
◇ w/o E Trace	62.89	52.75	61.18	58.94	↓3.30
◇ w/o All Parts	58.54	55.16	51.14	54.95	↓7.29
Mistral (7B)	**79.85**	**60.74**	**66.54**	**69.04**	-
◇ w P-Random	77.90	60.63	64.27	67.60	↓1.44
◇ w P-Input	78.79	60.47	62.80	67.35	↓1.69
◇ w/o Q Filter	78.28	60.64	58.85	65.92	↓3.12
◇ w/o P&Q Filter	76.60	60.42	60.14	65.72	↓3.32
◇ w/o E Trace	78.69	60.18	64.38	67.75	↓1.29
◇ w/o All Parts	74.62	59.11	59.27	64.33	↓4.71

Table 14: Ablation study for MPT (7B), Falcon (7B), and Mistral (7B) trained with AbsInstruct. Macro F1-scores are exhibited, and Δ_{All} indicates score changes.

D.4 Full Results of Ablation Study

Here, we present the full ablation study results of all LLMs trained with our framework AbsInstruct in Tables 14 and 15.

D.5 Study of Diversity Filter

In this appendix, we study the role of diversity filters in our framework AbsInstruct. Here, we remove the diversity filter and analyze the performance of the ablated framework.

First, we inspect the diversity of examples collected by the ablated framework. We compute the average ROUGE-L similarity between the head events and between explanation traces. From the Table 18, we can see that the average ROUGE-L similarities are no more than 0.2 for head events and 0.3 for explanation traces. Meanwhile, we also compute the proportion of unique head events and explanation traces based on ROUGE-L, following previous work (Wang et al., 2023b). A head event x is unique if $Rouge_L(C, x) \leq 0.7$, where C is other head events collected by our framework. We apply the same criterion to identify unique data for expla-

Models	Noun	Verb	Event	All	Δ_{All}
Llama2 (7B)	**75.81**	**59.07**	**68.00**	**67.63**	-
◇ w P-Random	69.56	58.48	66.04	64.69	↓2.94
◇ w P-Input	69.92	58.43	66.34	64.90	↓2.73
◇ w/o Q Filter	65.06	56.90	62.70	61.55	↓6.08
◇ w/o P&Q Filter	65.79	57.27	54.52	59.19	↓8.44
◇ w/o E Trace	69.98	58.25	66.27	64.84	↓2.79
◇ w/o All Parts	66.34	55.72	55.11	59.05	↓8.58
Llama2 (13B)	**80.35**	**60.58**	**67.24**	**69.39**	-
◇ w P-Random	69.73	60.19	59.40	63.11	↓6.28
◇ w P-Input	78.46	59.18	65.61	67.75	↓1.64
◇ w/o Q Filter	72.64	60.17	52.10	61.64	↓7.75
◇ w/o P&Q Filter	74.83	59.88	52.54	62.42	↓6.97
◇ w/o E Trace	79.88	60.46	65.46	68.60	↓0.79
◇ w/o All Parts	76.05	60.36	59.59	65.33	↓4.06

Table 15: Ablation study for Llama2 (7B) and Llama2 (13B) trained with ABSINSTRUCT. Macro F1-scores are exhibited, and Δ_{All} indicates score changes.

Models	Acc	Ma-F1	Δ_{Acc}	$\Delta_{\text{Ma-F1}}$
Fine-tuned on AbsPyramid				
MPT (7B)	60.42	60.27	-	-
Falcon (7B)	64.22	64.22	-	-
Mistral (7B)	64.81	64.78	-	-
Llama2 (7B)	62.40	62.13	-	-
Llama2 (13B)	64.28	64.25	-	-
Direct Injection				
MPT (7B)	63.97	54.35	↑3.55	↓5.92
Falcon (7B)	61.46	55.60	↓2.76	↓8.62
Mistral (7B)	70.81	65.26	↑6.00	↑0.48
Llama2 (7B)	69.87	65.92	↑7.47	↑3.79
Llama2 (13B)	72.35	68.52	↑8.07	↑4.27
AbsInstruct				
MPT (7B)	71.32	67.55	↑10.90	↑7.28
Falcon (7B)	67.82	65.94	↑3.60	↑1.72
Mistral (7B)	**78.21**	**76.65**	↑13.40	↑11.87
Llama2 (7B)	76.58	75.51	↑**14.18**	↑**13.38**
Llama2 (13B)	77.07	75.44	↑12.79	↑11.19

Table 16: The out-of-domain performance on the AbstractATOMIC dataset. Δ_{Acc} and $\Delta_{\text{Ma-F1}}$ mean improvements compared to LLMs fine-tuned on ABSPYRAMID.

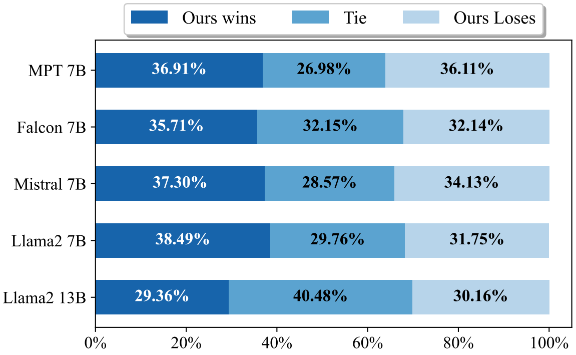

Figure 8: Human preference evaluation, comparing our framework to LLMs trained on Alpaca across 252 test prompts from SELF-INSTRUCT.

nation traces. From Table 18, we can see that more than 96% of head events and explanation traces are unique. These findings of average ROUGE-L and uniqueness percentages demonstrate that our dataset can collect quite diverse examples even without the diversity filter.

Then, we test the performance of the ABSINSTRUCT framework without the diversity filter, shown in Table 19. We can observe that the performance of all LLMs varies slightly. While we add a filter in our framework to guarantee the diversity of collected examples, our study verifies that the data collected by the ablated framework is already highly diverse.

D.6 Full Results of Out-of-Domain Evaluation

As we only plot the Macro F1-scores in Figure 5, we provide the full results on the AbstractATOMIC dataset across all metrics in Table 16. Meanwhile, we provide the results of all LLMs on the Levy/Holt dataset in Table 17.

D.7 Full Results of ChatGPT Rationales

As we only plot the Macro F1-score on the whole test set of ABSPYRAMID in Figure 6, we provide the full results on each entailment relation of ABSPYRAMID in Table 20.

D.8 Human Instruction Following

As previous works (Ouyang et al., 2022; Zhao et al., 2023) suggest a disparity between NLP tasks and human requests, we manually evaluate our framework on the 252 expert-curated instructions of SELF-INSTRUCT (Wang et al., 2023b) to better understand the alignment with human values. Similar to our evaluation on SuperNI, we consider LLMs trained on Alpaca as baselines. Three expert annotators are asked to compare responses from our framework to the baseline and label which one they prefer. We provide annotation details in Appendix B. Our human preference annotation results are plotted in Figure 8. We observe that a significant portion of prompts are labeled as “Tie.” Also, the winning rates appear to be comparable to, or even exceed, those of baselines preferred. These findings again manifest that our framework preserves LLMs’ general capabilities while enhancing their abstraction ability.

Models	Acc	Ma-F1	Δ_{Acc}	$\Delta_{\text{Ma-F1}}$
Fine-tuned on AbsPyramid				
MPT (7B)	80.38	71.47	-	-
Falcon (7B)	67.55	63.82	-	-
Mistral (7B)	79.32	72.66	-	-
Llama2 (7B)	78.69	71.07	-	-
Llama2 (13B)	82.11	71.25	-	-
Direct Injection				
MPT (7B)	78.79	56.69	↓1.59	↓14.78
Falcon (7B)	47.54	47.12	↓20.01	↓16.70
Mistral (7B)	85.34	74.55	↑6.02	↑1.89
Llama2 (7B)	84.29	74.00	↑5.60	↑2.93
Llama2 (13B)	85.51	76.27	↑3.40	↑5.02
AbsInstruct				
MPT (7B)	79.57	70.70	↓0.81	↓0.77
Falcon (7B)	76.19	69.97	**↑8.64**	↑6.15
Mistral (7B)	86.61	77.80	↑7.29	↑5.14
Llama2 (7B)	84.31	78.76	↑5.62	↑7.69
Llama2 (13B)	**87.11**	**79.89**	↑5.00	**↑8.64**

Table 17: The out-of-domain performance on the Levy/Holt dataset. Δ_{Acc} and $\Delta_{\text{Ma-F1}}$ mean improvements compared to LLMs fine-tuned on ABSPYRAMID.

Models	Head Event		Exp. Trace	
	Avg.	**Uni.**	**Avg.**	**Uni.**
MPT (7B)	0.164	96.00	0.272	96.17
Falcon (7B)	0.164	96.17	0.276	96.33
Mistral (7B)	0.157	96.33	0.258	96.83
Llama2 (7B)	0.161	96.17	0.261	96.83
Llama2 (13B)	0.161	96.00	0.256	97.00

Table 18: Analysis of diversity of examples collected by our framework when the diversity filter is removed. We list the average ROUGE-L similarity between every pair of samples and the percentage of unique examples.

D.9 Case Study and Error Analysis

In this section, we provide two examples with responses from our framework and the "Direct Injection" baseline. Here, the LLM we use is Llama2 (7B). As shown in Table 21, we can see that the baseline model cannot generate the correct answers. In contrast, our framework can explain and compare meanings of the instances and concepts in these examples and then give correct labels.

Then, we also provide an example in Table 22, where our framework (Llama2 7B) gives wrong predictions. We can see that the model makes wrong conclusions while it explains the instance and concept correctly.

Models	Noun	Verb	Event	All	Δ_{All}
MPT (7B)	70.27	58.40	64.04	64.24	↓0.65
Falcon (7B)	66.78	55.88	64.10	62.25	↑0.01
Mistral (7B)	80.05	60.78	67.08	69.31	↑0.27
Llama2 (7B)	74.35	59.44	67.27	67.02	↓0.61
Llama2 (13B)	80.49	60.51	66.92	69.31	↓0.08

Table 19: The performance of ablating the diversity filter. We only see fluctuations due to the high diversity of examples, even without the diversity filter.

Models	Noun	Verb	Event	All	Δ_{All}
MPT (7B)	70.73	58.64	65.74	65.04	↑0.15
Falcon (7B)	67.89	57.49	61.80	62.39	↑0.15
Mistral (7B)	76.52	60.93	67.90	68.45	↓0.59
Llama2 (7B)	74.22	59.36	66.84	66.81	↓0.82
Llama2 (13B)	78.48	59.79	68.14	68.80	↓0.59

Table 20: Macro F1-scores with ChatGPT as the source of explanation traces. Δ_{All} means score changes compared to that with GPT4 as the source.

E Study of Filtered Examples

In this appendix, we provide a few examples discarded by each quality filter to show their effectiveness. First, we provide two examples that are discarded by the prediction filter in Table 24. In these cases, GPT4 generates wrong explanations and predicts wrong labels for these examples. For example, GPT4 considers "marriage" as "a social and legal institution," while it refers to "the legal relationship between a husband and wife" in the current context. Meanwhile, GPT4 considers "concession" as a "grant or a contract," while it refers to "a right or an advantage that is given to a group of people, an organization, etc." We can find that GPT4 can generate wrong explanations and in turn, get wrong answers.

Then, we also list two examples discarded by the keyword filter in Table 25. While GPT4 predicts the correct labels for these examples, it fails to explain the correct instance or concepts. For example, GPT4 explains "dad-mouthing" in the first example instead of the correct instance "quoted." In the second example, GPT4 changes the original sentence to "the things that happen to PersonY," which is indeed a fragment and lacks a complete sentence structure.

Example #1
Entailment Relation: *Noun-Entail*, **Label:** 1 **Instance:** walls, **Concept:** architectural element
Head Event: The **walls** are white **Tail Event:** The **architectural elements** are white
Our Framework Response: Step 1: The original word "walls" refers to the structural elements that divide or enclose an area, typically made of a solid material such as brick, stone, or wood. Meanwhile, the new word "architectural element" refers to any part of a building that is designed and constructed by an architect or engineer, such as walls, windows, doors, stairs, and roofs. Step 2: Yes, the meaning of "architectural element" encompasses "walls." **Predicted Label:** 1
Baseline Response: No. **Predicted Label:** 0
Example #2
Entailment Relation: *Noun-Entail*, **Label:** 1 **Instance:** routine investigation controllers, **Concept:** controllers
Head Event: A **routine investigation controllers** are usually given any needed training. **Tail Event:** A **controllers** are usually given any needed training.
Our Framework Response: Step 1: The original word "routine investigation controllers" refers to individuals or entities that are responsible for conducting routine investigations. Meanwhile, the new word "controllers" refers to individuals or entities that have control over something. Step 2: Yes, the meaning of "controllers" encompasses "routine investigation controllers." **Predicted Label:** 1
Baseline Response: No. **Predicted Label**: 0

Table 21: Two examples demonstrating that our framework can deliver right answers with explanations, whereas the baseline "Direct Injection" falls short in providing correct answers. The LLM we use here is Llama2 (7B).

Example #1
Entailment Relation: *Verb-Entail*, **Label:** 0 **Instance:** remove, **Concept:** withdraw
Head Event: The asphalt shingles should be **removed** first. **Tail Event:** The asphalt shingles should be **withdrawed** first.
Our Framework Response: Step 1: The original word "removed" means to take something away or out of a place. Meanwhile, The new word "withdraw" means to take something back or away from a place. Step 2: Yes, the meaning of "withdraw" encompasses "removed." **Predicted Label:** 1

Table 22: An example showing some errors made by our framework. The LLM we use here is Llama2 (7B).

Methods	Backbone	Noun		Verb		Event		All	
		Acc	Ma-F1	Acc	Ma-F1	Acc	Ma-F1	Acc	Ma-F1
Random	-	50.00	49.67	50.00	49.97	50.00	49.01	50.00	49.55
Alpaca (10-shot)	MPT (7B)	44.70	35.68	49.23	37.94	65.53	44.75	53.16	39.45
	Falcon (7B)	59.98	54.61	55.60	55.45	63.75	45.02	59.77	51.69
	Mistral (7B)	74.81	73.19	58.76	58.20	65.69	58.78	66.42	63.39
	Llama2 (7B)	63.62	63.55	54.86	52.23	69.44	60.60	62.64	58.79
	Llama2 (13B)	75.13	72.41	58.68	58.66	66.97	61.99	66.93	64.35
Direct Injection	MPT (7B)	64.72	64.14	54.25	52.65	52.14	52.03	57.04	56.27
	Falcon (7B)	63.16	58.47	54.47	54.31	51.72	51.67	56.45	54.82
	Mistral (7B)	74.30	74.18	59.56	59.24	59.63	59.05	64.49	64.16
	Llama2 (7B)	67.30	66.57	55.24	54.26	57.51	57.49	60.02	59.44
	Llama2 (13B)	74.55	73.87	59.89	59.74	61.36	60.58	65.27	64.73
AbsInstruct	MPT (7B)	72.27	71.94	59.13	59.11	69.18	66.76	66.86	65.94
	Falcon (7B)	67.53	67.14	56.42	55.56	68.54	63.49	64.17	62.06
	Mistral (7B)	79.98	79.45	59.98	59.92	71.96	67.54	70.64	68.97
	Llama2 (7B)	77.68	76.69	60.06	59.72	**72.86**	**68.28**	70.20	68.23
	Llama2 (13B)	**80.33**	**79.73**	**60.25**	**60.24**	72.68	68.00	**71.09**	**69.32**

Table 23: Performance of AbsInstruct and baselines on the validation set of AbsPyramid. **Acc** and **Ma-F1** represent Accuracy and Macro F1-score, respectively. We bold the best score and underline the second-best score.

Example #1
Entailment Relation: *Noun-Entail* **Instance:** marriage, **Concept:** marriage bond
Head Event: Their **marriage** declared invalid. **Tail Event:** Their **marriage bond** declared invalid.
GPT4 Rationale: Step 1: The original word "marriage" broadly refers to a social and legal institution, which encompasses a wide range of practices, ceremonies, rights, and significances depending on the culture and context. Step 2: The new word "marriage bond" typically refers to the specific connection, commitment, or tie that exists between the married individuals. Step 3: No.
Ground Truth: Positive, **Prediction:** Negative.

Example #2
Entailment Relation: *Event-Entail* **Instance:** It gives PeopleX the right to fish, **Concept:** fishing concession
Head Event: It gives PeopleX the right to fish **Tail Event:** fishing concession
GPT4 Rationale: Step 1: The meaning of the sentence "It gives PeopleX the right to fish" is that a certain group, PeopleX, is granted the permission or legal authority to engage in fishing activities. Step 2: The meaning of the abstract description "fishing concession" generally refers to a grant or contract giving an individual or group the right to operate a fishing activity in a specified area or under certain conditions. Step 3: Yes.
Ground Truth: Negative, **Prediction:** Positive.

Table 24: Two examples filtered by the Prediction Filter. GPT4 might generate wrong explanations with wrong predicted labels.

Example #1
Entailment Relation: *Verb-Entail*, **Label:** Negative **Instance:** quoted, **Concept:** intercommunicated
Head Event: PersonX is **quoted** bad-mouthing teammates **Tail Event:** PersonX is **intercommunicated** bad-mouthing teammates
GPT4 Rationale: Step 1: "Bad-mouthing" means to speak negatively or disparagingly about someone. Step 2: "Intercommunicate" means to communicate with each other, implying a mutual exchange of information. Step 3: No.

Example #2
Entailment Relation: *Event-Entail*, **Label:** Positive **Instance:** The things happen to PersonY, **Concept:** life circumstances
Head Event: The things happen to PersonY **Tail Event:** life circumstances
GPT4 Rationale: Step 1: The sentence "the things that happen to PersonY" refers to any events, situations, or experiences that occur in the life of an individual named PersonY. Step 2: The term "life circumstances" broadly refers to the conditions or events that collectively define an individual's situation in life at any given time, encompassing a range of personal, social, economic, and environmental factors that affect the individual. Step 3: Yes.

Table 25: Two examples filtered by the Keyword Filter. GPT4 might generate explanations of wrong words and rewrite the instance or concept.