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

Zhaowei Wang¹, Wei Fan¹, Qing Zong¹, Hongming Zhang², Sehyun Choi¹, Tianqing Fang¹, Xin Liu¹, Yangqiu Song¹, Ginny Y. Wong¹, & Simon See⁴
¹Department of Computer Science and Engineering, HKUST
²Tencent AI Lab, Bellevue, USA, ³Amazon.com Inc, Palo Alto, USA
⁴NVIDIA AI Technology Center (NAITC), 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 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. (2023d) 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.

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...
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 probability 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., 2023d) 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. (2023d) 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 **typically** 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
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., 2023d), 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., 2023d). 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:

\[
\begin{align*}
\hat{y} &= \theta_{LLM}(i, x, e) \\
\mathbb{1}\{\hat{y} = y\} \\
\end{align*}
\]

where \(\theta_{LLM}\) 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 \(\mathbb{1}\{\hat{y} = y\}\) 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 pre-training, 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
the model’s knowledge. Concretely, the plausibility is computed as the normalized conditional probability of the response $r$ given the instruction $i$ and input $x$:

$$\text{Plausibility}(i, x, r) = P_{\theta}(r| i, x) \frac{1}{N},$$

(2)

where $\theta$ are the parameters of the pre-trained model, and $N$ is the number of tokens in $r$. The above equation is equivalent to the reciprocal of the perplexity of $r$ conditional on $i$ and $x$. 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). In practice, we take the logarithm of the above equation to ensure numerical stability.

### 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.

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.
5.2 Baseline Methods

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

**Direct Injection:** This baseline randomly samples tuples from ABSYRAMID 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).
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 “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, Llama2 (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 normalized conditional probabilities of explanation traces. Here, we compute the normalized 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 Table 3, 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 generalization 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.

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.

---

**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.

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

Abstraction knowledge is a critical kind of knowledge in human intelligence, as shown in previous cognitive studies (Minsky, 1980). Meanwhile, recent research (Zheng et al., 2023a; Gao et al., 2024) shows that using LLMs’ abstraction ability can better solve general NLP tasks, including STEM questions (Hendrycks et al., 2020; Miao et al., 2020), Knowledge QA (Yang et al., 2018; Kwiatkowski et al., 2019; Joshi et al., 2017), and Multi-Hop Reasoning (Trivedi et al., 2022; Geva et al., 2021). Given this evidence, we can see that abstraction is also essential for a broad range of complex language understanding and reasoning tasks.

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. In the experiments, we compare our framework with a lot of strong baselines to demonstrate the framework’s effectiveness. We also provide comprehensive ablation studies of our framework and show its effectiveness on two out-of-domain datasets. What’s more, evaluations on instruction datasets also show that our framework can improve the abstraction ability of LLMs without sacrificing LLMs’ general instruction following ability.

For future work, we can study how to equip LLMs with more abstraction knowledge during pre-training. More importantly, we also leave the study of using this enhanced abstraction knowledge in downstream tasks as future work.

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., 2023b; Wu et al., 2024), event-centric knowledge (Wang et al., 2023c, 2022c; Gao et al., 2023a; Do et al., 2024), intention detection (Wu et al., 2023), and knowledge population (Shen et al., 2023). Meanwhile, we can extend our abstraction knowledge to multimodal, like exploring knowledge from given images (Shen et al., 2024; Cui et al., 2024).

Ethics Statement

We evaluate the abstraction ability on ABSPYRAMID (Wang et al., 2023d), 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. The instruction datasets SuperNI (Wang et al., 2022b) and SELF-INSTRUCT (Wang et al., 2023b) are released under the Apache-2.0 License. Meanwhile, the Alpaca (Taori et al., 2023) dataset is released under the CC BY NC 4.0 License.

Human evaluations are performed by three expert annotators with at least one year of expertise in NLP to ensure 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 AI Lab Rhino-bird Focused Research Program. We also thank the support from NVIDIA AI Technology Center (NV AITC) and the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).
References


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


Hejie Cui, Xinyu Fang, Zihan Zhang, Ran Xu, Xuan Kan, Xin Liu, Yue Yu, Manling Li, Yangqiu Song, and Carl Yang. 2024. Open visual knowledge extraction via relation-oriented multimodality model prompting. Advances in Neural Information Processing Systems, 36.


Peter Hase, Mohit Bansal, Been Kim, and Asma Ghan- 
deharioun. 2023. Does localization inform editing? 
surprising differences in causality-based localization 

Mutian He, Tianqing Fang, Weiqi Wang, and Yangqiu Song. 2022. Acquiring and modelling abstract com-

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, 
Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 
In International Conference on Learning 
Representations.

Xavier Ricketts Holt. 2018. Probabilistic Models of 
Relational Implication. Ph.D. thesis, Macquarie Uni-
versity.

Mohammad Javad Hosseini, Nathanael Chambers, Siva 
Reddy, Xavier R Holt, Shay B Cohen, Mark Johnson, 
and Mark Steedman. 2018. Learning typed entail-

Mohammad Javad Hosseini, Shay B Cohen, Mark John-
son, and Mark Steedman. 2019. Duality of link pre-

Mohammad Javad Hosseini, Shay B Cohen, Mark John-
son, and Mark Steedman. 2021. Open-domain con-
textual link prediction and its complementarity with 

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, 
Wangchunshu Zhou, Guangtao Zeng, Antoine Bosse-
lut, and Mrinmayra Sachan. 2023. Towards a me-
chanistic interpretation of multi-step reasoning capa-

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 lan-
guage models. In International Conference on Learning 
Representations.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing 
Huang. 2019. Glossbert: Bert for word sense disam-
Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3509–3514.

Gatri Asti Putri Indarti. 2015. Distinguishing entailment 

Aditi Jha, Sam Havens, Jeremy Dohmann, Alex Trott, 
and Jacob Portes. 2023. Limit: Less is more for in-

Albert Q Jiang, Alexandre Sablayrolles, Arthur Men-
sch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guilla-

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke 
Zettlemoyer. 2017. Triviaqa: A large scale distantly 
supervised challenge dataset for reading comprehen-
sion. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Vol-
ume 1: Long Papers), pages 1601–1611.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science 

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

Nikita Kitaev and Dan Klein. 2018. Constituency pars-

Po-Nien Kung and Nanyun Peng. 2023. Do models 
really learn to follow instructions? an empirical 

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Red-
field, Michael Collins, Ankur Parikh, Chris Alberti, 
Danielle Epstein, Illia Polosukhin, Jacob Devlin, Ken-

Omer Levy and Ido Dagan. 2016. Annotating rela-
tion inference in context via question answering. In 

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio 
Petroni, Vladimir Karpukhin, Naman Goyal, Hein-
rich Küttler, Mike Lewis, Wen-tau Yih, Tim Rock-

evaluation of summaries. In Text summarization 
branches out, pages 74–81.

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


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? 


A A BS INSTRUCT 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 ABS PYRAMID 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.

<table>
<thead>
<tr>
<th>Table 6: The concrete prompts we used in our A</th>
<th>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. Also, &lt;ins mean&gt; and &lt;cpt mean&gt; will be replaced with the meanings of real instances and concepts.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noun-Entail Input:</strong> In the sentence [head], does the meaning of [cpt] encompass [ins]?</td>
<td><strong>Noun-Entail:</strong> 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]?</td>
</tr>
</tbody>
</table>
| **Verb-Entail Input:** In the sentence [head], does the meaning of [cpt] encompass [ins]? | **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]?
| **Event-Entail Input:** Can we consider [cpt] as an abstract description of the sentence [head]?
| **Event-Entail:** Identify abstract descriptions of specific sentences. Can we consider [cpt] as an abstract description of the sentence [head]?

(c) Response templates used by our framework.

**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 Alpaca 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 \( \alpha \) 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. 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.

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. Hyponyms 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. Hyponyms 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.
**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.

<table>
<thead>
<tr>
<th>Exemplars and test example:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. In the sentence [head]^[1], is [cpt]^[1] a hypernym of [ins]^[1]? Yes, (No.)</td>
<td></td>
</tr>
<tr>
<td>2. In the sentence [head]^[2], is [cpt]^[2] a hypernym of [ins]^[2]? Yes, (No.)</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
</tr>
<tr>
<td>11. In the sentence [head]^[11], is [cpt]^[11] a hypernym of [ins]^[11]?</td>
<td></td>
</tr>
</tbody>
</table>
We collect the performance of our framework ABSINSTRUCT and baselines on the validation set of the ABSPYRAMID in Table 23.

### 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 $\text{Rouge}_L(C, x) \leq 0.7$, where $C$ is other head events collected by our framework. We apply the same criterion to identify unique data for explanation traces.

In Table 19, we can see that the diversity filter is very effective. The average ROUGE-L similarity between head events is reduced from 0.21 to 0.02, and the average ROUGE-L similarity between explanation traces is reduced from 0.31 to 0.10. Moreover, the proportion of unique head events is increased from 0.02 to 0.18, and the proportion of unique explanation traces is increased from 0.00 to 0.09.

### Table 19: Ablation Study Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Noun</th>
<th>Verb</th>
<th>Event</th>
<th>All</th>
<th>$\Delta_{\text{All}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPT (7B)</td>
<td>70.89</td>
<td>58.63</td>
<td>65.16</td>
<td>64.89</td>
<td>-</td>
</tr>
<tr>
<td>w/o All Parts</td>
<td>77.90</td>
<td>60.63</td>
<td>64.27</td>
<td>67.60</td>
<td>1.14</td>
</tr>
<tr>
<td>w/o P-Random</td>
<td>59.75</td>
<td>53.92</td>
<td>58.95</td>
<td>57.60</td>
<td>1.19</td>
</tr>
<tr>
<td>w/o Q Filter</td>
<td>78.28</td>
<td>60.42</td>
<td>64.14</td>
<td>65.72</td>
<td>1.32</td>
</tr>
<tr>
<td>w/o E Trace</td>
<td>78.69</td>
<td>60.18</td>
<td>64.38</td>
<td>67.75</td>
<td>1.29</td>
</tr>
<tr>
<td>w/o All Parts</td>
<td>74.62</td>
<td>59.11</td>
<td>59.27</td>
<td>64.33</td>
<td>1.71</td>
</tr>
</tbody>
</table>

Table 19: Ablation study for MPT (7B), Falcon (7B), and Mistral (7B) trained with ABSINSTRUCT. Macro F1-scores are exhibited, and $\Delta_{\text{All}}$ indicates score changes.
<table>
<thead>
<tr>
<th>Models</th>
<th>Noun</th>
<th>Verb</th>
<th>Event</th>
<th>All</th>
<th>∆All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llama2 (7B)</td>
<td>75.81</td>
<td>59.07</td>
<td>68.00</td>
<td>67.63</td>
<td>-</td>
</tr>
<tr>
<td>-w P-Random</td>
<td>69.56</td>
<td>58.48</td>
<td>66.04</td>
<td>64.69</td>
<td></td>
</tr>
<tr>
<td>-w P-Input</td>
<td>69.92</td>
<td>58.43</td>
<td>66.34</td>
<td>64.90</td>
<td></td>
</tr>
<tr>
<td>-w/o Q Filter</td>
<td>65.06</td>
<td>56.90</td>
<td>62.70</td>
<td>61.55</td>
<td></td>
</tr>
<tr>
<td>-w/o P&amp;Q Filter</td>
<td>65.79</td>
<td>57.27</td>
<td>54.52</td>
<td>59.19</td>
<td></td>
</tr>
<tr>
<td>-w/o E Trace</td>
<td>69.98</td>
<td>58.25</td>
<td>66.27</td>
<td>64.84</td>
<td></td>
</tr>
<tr>
<td>-w/o All Parts</td>
<td>66.34</td>
<td>55.72</td>
<td>55.11</td>
<td>59.05</td>
<td></td>
</tr>
<tr>
<td>Llama2 (13B)</td>
<td>80.35</td>
<td>60.58</td>
<td>67.24</td>
<td>69.39</td>
<td>-</td>
</tr>
<tr>
<td>-w P-Random</td>
<td>78.46</td>
<td>60.17</td>
<td>52.10</td>
<td>61.64</td>
<td>-8.24</td>
</tr>
<tr>
<td>-w P-Input</td>
<td>76.71</td>
<td>59.85</td>
<td>54.47</td>
<td>61.19</td>
<td>-7.37</td>
</tr>
<tr>
<td>-w/o Q Filter</td>
<td>72.64</td>
<td>60.17</td>
<td>52.10</td>
<td>61.64</td>
<td>-7.75</td>
</tr>
<tr>
<td>-w/o P&amp;Q Filter</td>
<td>74.83</td>
<td>59.88</td>
<td>52.54</td>
<td>62.42</td>
<td>-6.95</td>
</tr>
<tr>
<td>-w/o E Trace</td>
<td>79.88</td>
<td>60.46</td>
<td>65.46</td>
<td>68.60</td>
<td>-0.79</td>
</tr>
<tr>
<td>-w/o All Parts</td>
<td>76.05</td>
<td>60.36</td>
<td>59.59</td>
<td>65.33</td>
<td>-4.06</td>
</tr>
</tbody>
</table>

Table 15: Ablation study for Llama2 (7B) and Llama2 (13B) trained with ABSTRACT-STRUCT. Macro F1-scores are exhibited, and ∆All indicates score changes.

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 ABSTRACT-STRUCT 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.
Table 17: The out-of-domain performance on the Levy/Holt dataset. ∆Acc and ∆Ma-F1 mean improvements compared to LLMs fine-tuned on ABSPYRAMID.

<table>
<thead>
<tr>
<th>Models</th>
<th>Acc</th>
<th>Ma-F1</th>
<th>∆Acc</th>
<th>∆Ma-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-tuned on AbsPyramid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPT (7B)</td>
<td>80.38</td>
<td>71.47</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Falcon (7B)</td>
<td>67.55</td>
<td>63.82</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mistral (7B)</td>
<td>79.32</td>
<td>72.66</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Llama2 (7B)</td>
<td>78.69</td>
<td>71.07</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Llama2 (13B)</td>
<td>82.11</td>
<td>71.25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Direct Injection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPT (7B)</td>
<td>78.79</td>
<td>56.69</td>
<td>↓1.59</td>
<td>↓14.78</td>
</tr>
<tr>
<td>Falcon (7B)</td>
<td>47.54</td>
<td>47.12</td>
<td>↓20.01</td>
<td>↓16.70</td>
</tr>
<tr>
<td>Mistral (7B)</td>
<td>85.34</td>
<td>74.55</td>
<td>↑1.89</td>
<td></td>
</tr>
<tr>
<td>Llama2 (7B)</td>
<td>84.29</td>
<td>74.00</td>
<td>↑5.60</td>
<td>↑2.93</td>
</tr>
<tr>
<td>Llama2 (13B)</td>
<td>85.51</td>
<td>76.27</td>
<td>↑3.40</td>
<td>↑5.02</td>
</tr>
<tr>
<td>AbsInstruct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPT (7B)</td>
<td>79.57</td>
<td>70.70</td>
<td>↓0.81</td>
<td>↓0.77</td>
</tr>
<tr>
<td>Falcon (7B)</td>
<td>76.19</td>
<td>69.97</td>
<td>↓8.64</td>
<td>↓6.15</td>
</tr>
<tr>
<td>Mistral (7B)</td>
<td>86.61</td>
<td>77.80</td>
<td>↑7.29</td>
<td>↑5.14</td>
</tr>
<tr>
<td>Llama2 (7B)</td>
<td>84.31</td>
<td>78.76</td>
<td>↑5.62</td>
<td>↑7.69</td>
</tr>
<tr>
<td>Llama2 (13B)</td>
<td>87.11</td>
<td>79.89</td>
<td>↑5.00</td>
<td>↑8.64</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Models</th>
<th>Head Event</th>
<th>Exp. Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPT (7B)</td>
<td>0.164</td>
<td>96.00</td>
</tr>
<tr>
<td>Falcon (7B)</td>
<td>0.164</td>
<td>96.17</td>
</tr>
<tr>
<td>Mistral (7B)</td>
<td>0.167</td>
<td>96.33</td>
</tr>
<tr>
<td>Llama2 (7B)</td>
<td>0.161</td>
<td>96.17</td>
</tr>
<tr>
<td>Llama2 (13B)</td>
<td>0.161</td>
<td>96.00</td>
</tr>
</tbody>
</table>

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.

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 withdrew 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).
<table>
<thead>
<tr>
<th>Methods</th>
<th>Backbone</th>
<th>Noun Acc</th>
<th>Verb Acc</th>
<th>Event Acc</th>
<th>All Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ma-F1</td>
<td>Ma-F1</td>
<td>Ma-F1</td>
<td>Ma-F1</td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td>50.00</td>
<td>49.67</td>
<td>50.00</td>
<td>49.97</td>
</tr>
<tr>
<td>Alpaca (10-shot)</td>
<td></td>
<td>44.70</td>
<td>35.68</td>
<td>49.23</td>
<td>37.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59.98</td>
<td>54.61</td>
<td>55.60</td>
<td>55.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.81</td>
<td>73.19</td>
<td>58.76</td>
<td>58.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.62</td>
<td>63.55</td>
<td>54.86</td>
<td>52.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.13</td>
<td>72.41</td>
<td>58.68</td>
<td>58.66</td>
</tr>
<tr>
<td>Direct Injection</td>
<td></td>
<td>64.72</td>
<td>64.14</td>
<td>54.25</td>
<td>52.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.16</td>
<td>58.47</td>
<td>54.47</td>
<td>54.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.30</td>
<td>74.18</td>
<td>59.56</td>
<td>59.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>67.30</td>
<td>66.57</td>
<td>55.24</td>
<td>54.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.55</td>
<td>73.87</td>
<td>59.89</td>
<td>59.74</td>
</tr>
<tr>
<td>AbsInstruct</td>
<td></td>
<td>72.27</td>
<td>71.94</td>
<td>59.13</td>
<td>59.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>67.53</td>
<td>67.14</td>
<td>56.42</td>
<td>55.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.98</td>
<td>79.45</td>
<td>59.98</td>
<td>59.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>77.68</td>
<td>76.69</td>
<td>60.06</td>
<td>59.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80.33</td>
<td>78.03</td>
<td>60.25</td>
<td>60.24</td>
</tr>
</tbody>
</table>

Table 23: Performance of ABSINSTRUCT and baselines on the validation set of ABS-PYRAMID. **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 term “fishing concession” generally refers to a permit or contract giving an individual 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.