AbsPyramid: Benchmarking the Abstraction Ability of Language Models with a Unified Entailment Graph

Anonymous ACL submission

Abstract

Cognitive research indicates that abstraction ability is essential in human intelligence, which remains under-explored in language models. In this paper, we present AbsPyramid, a unified entailment graph of 221K textual descriptions of abstraction knowledge. While existing resources only touch nouns or verbs within simplified events or specific domains, AbsPyramid collects abstract knowledge for three components of diverse events to comprehensively evaluate the abstraction ability of language models in the open domain. Experimental results demonstrate that current LLMs face challenges comprehending abstraction knowledge in zero-shot and few-shot settings. By training on our rich abstraction knowledge, we find LLMs can acquire basic abstraction abilities and generalize to unseen events. In the meantime, we empirically show that our benchmark is comprehensive to enhance LLMs across two previous abstraction tasks.

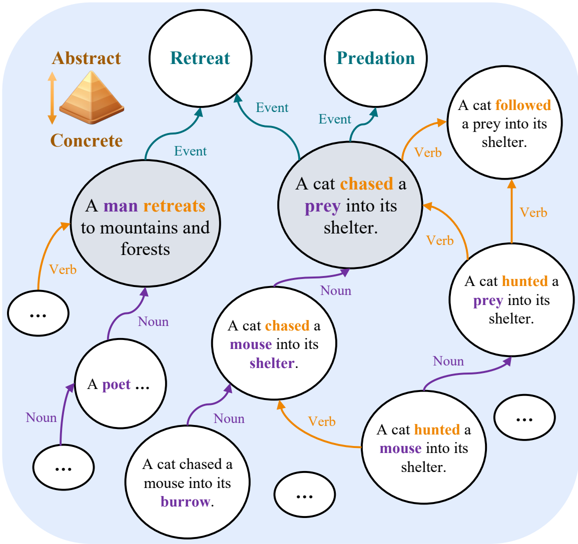


Figure 1: An illustration of our AbsPyramid benchmark. We identify three components of events (i.e., ***Noun***, ***Verb***, and ***Event*** as a whole) and collect abstract concepts entailed by them.

1 Introduction

Abstraction is about finding common properties among different things and forming a broader concept, like the concept "furniture" subsuming "sofa" and "table," a key dimension of human cognition (Colung and Smith, 2003; Russell and Norvig, 2010). With this ability, we can smoothly handle daily situations by learning from past experiences and generalizing to new circumstances (Saitta and Zucker, 2013). Substantively, Minsky (1980), in his *K-Theory*, suggested that our minds organize past experiences in a hierarchical pyramid, with higher parts corresponding to greater abstraction.

The NLP community has recently explored diverse, impressive abilities of LLMs, such as in-context learning (Brown et al., 2020), multi-step reasoning (Wei et al., 2022b), and instruction following (Sanh et al., 2022). Meanwhile, the ability to abstract, a core dimension of human cognition, has received less attention in the studies of LLMs.

Although sporadic works about abstraction knowledge exist, they focus solely on nouns or verbs within simplified events or specific domains, failing to consider a broader picture of abstraction. One category of works is building an entailment graph of verbs, first proposed by Berant et al. (2011) with several techniques to enhance it in the following works (Hosseini et al., 2018; McKenna et al., 2023). Those works consider events as a *verb* with two arguments (i.e., *subject* and *object*) and limit arguments to dozens of entity types to alleviate their graphs' sparsity issue. However, those simplifications considerably sacrifice the precise semantics of events. For example, the event "a cat chased a mouse into its burrow" in Figure 1 will be simplified into a tuple (animal, chase, animal), losing track of specific details of animals and location. Other than verbs, He et al. (2022) annotated an abstraction dataset, AbstractATOMIC, about entities and events using the Probase taxonomy (Wu et al., 2012). While their work curated thousands of abstract concepts, it is limited to the social commonsense domain as base events are sampled from ATOMIC (Sap et al., 2019).

Inspired by the cognitive study of ***abs***traction in the ***pyramid***-like hierarchy of human experiences (Minsky, 1980), we present ABSPYRAMID, a unified entailment graph to comprehensively evaluate language models' abstraction ability. We curated abstract concepts entailed by each of the three components of an event: nouns, verbs, and the event as a whole, unifying scopes and domains of all prior datasets. Specifically, we sample base events in textual descriptions from ASER (Zhang et al., 2020, 2022), an open-domain large-scale eventuality graph. We design heuristic rules to identify nouns and verbs from events and collect abstract concepts with WordNet (Miller, 1995) and LLMs prompting. Those concept candidates are then crowdsourced for validity, resulting in a graph of 221K examples. Compared with verb entailment graphs (Berant et al., 2011), ABSPYRAMID retains specific and accurate semantics of base events. Our benchmark features a diverse array of syntactic roles for real arguments instead of relying on (*subject*, *verb*, *object*) tuples with entity types. In contrast to AbstractATOMIC (He et al., 2022), our benchmark covers abstraction knowledge beyond the social commonsense thanks to the open domain corpora used in ASER. Also, we use LLMs to broaden collected abstract concepts, complementing the coverage of taxonomies.

On the ABSPYRAMID benchmark, we investigate whether LLMs can (1) identify valid abstract concepts and (2) generate abstract concepts. The evaluation results on 26 popular language models reveal that: (1) LLMs encounter difficulties understanding abstraction knowledge under both zero-shot and in-context learning settings. (2) In contrast, fine-tuned language models perform better at comprehending abstraction knowledge, especially for nouns. (3) Our benchmark incorporates comprehensive abstraction knowledge, which can improve LLMs' performance significantly across verb entailment graphs and AbstractATOMIC. To the best of our knowledge, ABSPYRAMID presents the first comprehensive evaluation of LLMs' abstraction ability. Our benchmark and experiment results provide valuable insights into the abstraction ability of language models and the progress of artificial intelligence within LLM.

2 Related Work

While the NLP community has studied various abilities of LLMs (Wei et al., 2022a; Chowdhery et al., 2023; Ouyang et al., 2022; Chung et al., 2022; Zhou et al., 2023), the abstraction ability of LLMs remains insufficiently studied. Unlike existing works that focus on entity-level abstraction (Clark et al., 2000; Van Durme et al., 2009; Song et al., 2011, 2015; Gong et al., 2016), our research delves into event-level abstraction with only a few works investigating some restricted aspects:

Verb Entailment Graph: Berant et al. (2011) first proposed the task of entailment graph construction of verbs. Following their work, various methods have been proposed to build better verb entailment graphs (Hosseini et al., 2018, 2019, 2021; Guillou et al., 2020; Chen et al., 2022; Li et al., 2022; McKenna et al., 2021, 2023). Nonetheless, those works consider verbs as binary relations with two arguments from a small set of entity types (e.g., 49 types in FIGER (Hosseini et al., 2018)), distorting the original semantics.

AbstractATOMIC: He et al. (2022) presented an annotated abstraction dataset. They recognized entities in head events from ATOMIC (Sap et al., 2019) and crowdsourced abstract concepts from the Probase taxonomy (Wu et al., 2012) for recognized entities and head events. Even though they compiled a dataset comprising thousands of examples, it is specific to the social commonsense domain due to the base events sampled from ATOMIC.

Textual and Linguistic Entailment: Besides the entailment between verbs, recognizing textual entailment has long been a vital task in the realm of NLP (Cooper et al., 1996; Dagan et al., 2005), also known as *natural language inference* (NLI). Researchers have built many large-scale datasets of NLI (Conneau et al., 2018; Williams et al., 2018; Nie et al., 2020) and its variants (Wang et al., 2019; Dalvi et al., 2021; Chen et al., 2023).

While similar to our task, textual entailment employs a relaxed definition of whether a human reader would ***typically infer*** a *hypothesis* from a given *premise* (MacCartney et al., 2007; Korman et al., 2018) instead of abstraction of the *premise*. For example, in SNLI (Bowman et al., 2015), we can infer *a boy is holding his arms out* from the premise *a boy looks down and spreads his arms wide* without any abstraction involved. In contrast,

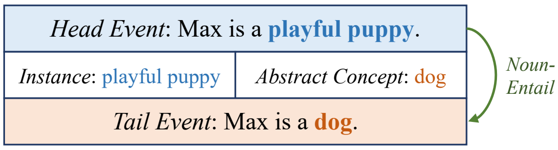


Figure 2: An illustration of the structure of abstraction knowledge, where ***entailment relation*** is *Noun-Entail*.

our work follows the definition of linguistic entailment (Beth, 1955), which arises from the semantics of linguistic expressions and is enforced by lexical meanings plus the laws of logic (Murphy, 2010; Sauerland and Stateva, 2007). For instance, *Max is a playful puppy* entails *Max is a dog* since one cannot be a playful puppy without being a dog.

3 Abstraction Knowledge Structure

ABSPYRAMID represents a large-scale abstraction repository of events in textual descriptions. This unified entailment graph contains 221K five-element tuples with the format of (***head event***, ***entailment relation***, ***tail event***, ***instance***, ***abstract concept***). In each tuple, we identify an ***instance*** in the ***head event*** and collect an ***abstract concept*** for it. Particularly, instances are identified from three components of the head event: nouns, verbs, and head event as a whole. Then, we replace the instance with its abstract concept to construct the ***tail event***, resulting in the tail event being linguistically entailed by the head event. According to three kinds of instances, we define three types of ***entailment relation***: *Noun-Entail*, *Verb-Entail*, and *Event-Entail*. We elaborate on each tuple element with a concrete example in Figure 2.

4 Data Curation Pipeline

To build ABSPYRAMID, we create a crowdsourcing framework that allows for a scalable, broad collection of abstraction knowledge in the above-mentioned format.

4.1 Compiling Head Events

We randomly sample 17K base eventualities from ASER as head events. Since ASER is an automatically extracted graph, some noisy extraction results may affect the quality of our benchmark. Thus, we design elaborate rules to clean ASER using lexical and dependency parsing features (Details in Appendix A.1). Meanwhile, ASER is extracted from six open domain corpora spanning Wikipedia[1], NYT (Sandhaus, 2008), Yelp[2], Reddit[3], etc. We only sample eventualities from NYT and Wikipedia due to the less formal nature of other corpora, such as diverse styles of comments on Yelp. To collect more general events, we replace tokens referring to people with a `Person` variable (e.g., replace I/we/she/... with `PersonX/Y/Z`), following previous work (Sap et al., 2019).

4.2 Identifying Instances

As mentioned earlier, our benchmark defines three entailment relations. For *Event-Entail*, we can directly use head events as identified instances. More intricately, we need to identify nouns and verbs as instances within head events when dealing with *Noun-Entail* and *Verb-Entail*. We design an algorithm to heuristically match nouns and verbs based on parsing results (e.g., POS-tags) provided by ASER (Details in Appendix A.2).

4.3 Collecting Abstract Concepts

Then, we collect abstract concepts for those identified instances through two methods: (1) retrieving from non-contextualized taxonomy and (2) prompting LLMs to generate candidates in free form.

Pilot Study: There are two taxonomies of words containing abstract concepts: WordNet (Miller, 1995) and Probase (Wu et al., 2012). WordNet contains hypernym relations, words with a broad meaning that more specific words (i.e., hyponyms) fall under. Probase automatically extracts instance-concept relations of nouns from corpora. Both aggregate all senses of each word without context.

Our pilot study reveals that WordNet effectively covers more than 90% of verbs within head events. Nonetheless, the coverage of nouns is unsatisfactory, as we can build a gigantic space of nominal phrases by adding modifiers. For example, we can easily form numerous phrases of "dog" by adding "guard," "hunting," or "white," etc. Our pilot study finds that only 6.3% of nominal phrases in head events are covered by WordNet. Likewise, the coverage of Probase is also unacceptable (29.6%).

Abstract Concepts for Nouns: Due to the limited coverage of nouns in taxonomies, we collect hypernyms for nouns by prompting an LLM. In detail, we prompt ChatGPT under the in-context

[1]https://dumps.wikimedia.org/enwiki
[2]https://www.yelp.com/dataset/challenge
[3]https://www.reddit.com/r/datasets/comments/3bxlg7

learning setting with the standard task-instruction-then-exemplar prompts (West et al., 2022):

<INSTRUCTION>
<EX$_1$-IN><EX$_1^{(1)}$-OUT> . . . <EX$_1^{(K)}$-OUT>
...
<EX$_N$-IN><EX$_N^{(1)}$-OUT> . . . <EX$_N^{(K)}$-OUT>
<EX$_{N+1}$-IN>

where **<INSTRUCTION>** describes the task of finding abstract concepts of a noun in our case. The input **<EX$_i$-IN>** is a head event with an identified noun, with output **<EX$_i^{(k)}$-OUT>** being an abstract concept. Given such a prompt, ChatGPT compactly generates K abstract concepts for each testing input. In the meantime, we design another prompt to elicit challenging negative examples that are highly related but not abstract concepts, such as "stream course" for "stream" in "the stream creates a peaceful ambiance." Prompts are shown in Appendix A.3 concretely, with N and K equal to 10.

Abstract Concepts for Verbs: We collect abstract concepts for verbs using hypernyms from WordNet, as verbs are well covered. We link verbs into WordNet and employ GlossBERT (Huang et al., 2019), a word-sense disambiguation (WSD) model, to select each verb's correct (at least most probable) word sense. Then, hypernyms of the correct word sense are collected as abstract concepts.

Abstract Concepts for Events: Events are more complex than nouns and verbs without relevant taxonomy. Thus, we again prompt ChatGPT to collect phrasal abstract concepts of each head event. We use the prompts similar to nouns with slight changes in verbalizing input tuples (More details in Appendix A.3). N and K are equal to 10.

4.4 Dataset Annotation

The last step of our data curation pipeline is to verify the validity of automatically collected abstract concepts. We create an annotation task for each entailment relation on Amazon Mechanical Turk (MTurk). In those tasks, we first give annotators detailed instructions about the validity of abstract concepts, like explanations of hypernyms. We provide annotators with five-element tuples, as mentioned in Section 3, asking them whether each abstract concept is valid. For *Verb-Entail*, we also provided meanings of each verb from WordNet for better understanding. Meanwhile, to ensure annotation quality, we introduce two qualification tests and two rounds of annotation refinement. Details of quality control and annotation agreements are shown in Appendix A.4.

REL.	# Total	# Train	# Valid	# Test	% Pos
NOUN	98,783	79,034	9,874	9,875	58.98
VERB	59,542	47,669	5,939	5,934	52.29
EVENT	62,472	49,988	6,237	6,247	64.77
ALL	220,797	176,691	22,050	22,056	58.82

Table 1: Statistics of ABSPYRAMID. **Pos** denotes positive rates. **REL.** indicates entailment relations. We split data into training, validation, and test sets (80:10:10).

5 ABSPYRAMID Overview

In this section, we carry out a thorough analysis of our benchmark ABSPYRAMID.

5.1 Benchmark Statistics

ABSPYRAMID is a large-scale benchmark comprising about 221K abstraction examples. Specific details are shown in Table 1. For breakdown details, we collected more than 98K, 59K, and 62K tuples for *Noun-Entail*, *Verb-Entail*, and *Event-Entail*. To better understand our benchmark, We compare it with the Levy/Holt dataset (Levy and Dagan, 2016; Holt, 2018), a dataset heavily used to evaluate verb entailment graphs, and AbstractATOMIC (He et al., 2022). Four statistical metrics are computed for multi-dimensional comparison, including data size, vocabulary size, percentage of unique abstract concepts, and social domain proportions, with results as follows.

Previous studies show that content generated by LMs, ChatGPT in our case, might lack diversity (Welleck et al., 2019). From Table 2, we can find that our benchmark has a much larger **data size** and **vocabulary size** than previous resources, showing the lexical diversity of our benchmark. In particular, the vocabulary size is more than three times that of prior resources.

We also compute the **percentage of unique abstract concepts** based on BLEU soft uniqueness (Zhu et al., 2018; West et al., 2022). An abstract concept x is unique if $BLEU_1(C, x) \leq 0.5$, where C is all concepts that share the same head event and identified instance with x, and 0.5 is an empirical threshold. Our benchmark has a percentage on par with other datasets, showing the efficacy of our data curation pipeline. Last, we also report the **social domain proportions**, where we count head events with `Person` variables. As shown in Table 2, all head events in AbstractATOMIC con-

Dataset	Data (K)	Vocab. (K)	Unique	Social
NOUN	98.78	20.95	93.57	19.88
VERB	59.54	11.86	95.74	40.02
EVENT	62.47	19.04	73.43	36.15
ALL	220.80	29.42	88.26	32.19
AbsAtomic	92.23	8.99	89.42	100.00
Levy/Holt	18.41	5.62	87.85	38.17

Table 2: Dataset comparison. Data size, vocabulary size, percentage of unique abstract concepts, and social domain proportion are listed.

REL.	# Total	# Train	# Valid	# Test	Avg-Ref
NOUN	58,266	52,440	2,910	2,916	5.58
VERB	31,132	28,018	1,556	1,558	2.90
EVENT	40,466	36,446	2,006	2,014	4.57
ALL	129,864	116,904	6,472	6,488	4.33

Table 3: The statistics of generation data. **Avg-Ref** means the average references per identified instance. **REL.** stands for entailment relations. Tuples are split into training, validation, and test sets (90:5:5).

tain `Person` variables since they are sampled from ATOMIC. In contrast, 32.19% of head events in ABSPYRAMID pertain to daily life experiences.

5.2 Evaluation Tasks

We study two tasks on our benchmark, abstraction detection and generation, to evaluate whether LLMs can detect and generate abstraction knowledge. In the detection task, models are given a five-element tuple (in Section 3) and are asked to decide if the abstract concept is valid. We split collected abstraction knowledge into training, validation, and test sets (80:10:10) to form the ABSPYRAMID$_{[\text{DET}]}$ dataset (in Table 1). In the generation task, models are requested to generate abstract concepts for a given tuple. We remove tuples with invalid abstract concepts and form ABSPYRAMID$_{[\text{GEN}]}$ dataset in Table 3. We ensure that tuples sharing the same head event and identified instances are in the same set for both datasets.

6 Abstraction Detection Experiment

In this section, we conduct extensive experiments on the ABSPYRAMID$_{[\text{DET}]}$ dataset to evaluate an abundance of language models and provide comprehensive analyses.

6.1 Experiment Setup

Evaluation Metric: We calculate Accuracy, Macro F1-score, and ROC-AUC between predicted and ground truth labels to evaluate all models.

Models We evaluate four categories of LMs. **(1) PLM + FT:** We fine-tune pre-trained LMs: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2020), in the base and large sizes. **(2) NLI + Zero&FT:** We include four models fine-tuned on NLI data: BART-large-mnli (Lewis et al., 2020a), RoBERTa-base/large-mnli (Liu et al., 2019), and DeBERTa-large-mnli (He et al., 2020). We assess the zero-shot capability of those models and fine-tune them on our dataset. **(3) LLM + LoRA:** We fine-tune representative LLMs with LoRA (Hu et al., 2021): Llama2 (7B, 13B) and Llama2-Chat (7B, 13B) (Touvron et al., 2023), Falcon (7B) and Falcon-Instruct (7B) (Penedo et al., 2023), and Mistral (7B) and Mistral-Instruct (7B) (Jiang et al., 2023). **(4) LLM API:** We assess a series of closed-source LLMs under the zero-shot and in-context learning setups, covering GPT3.5 (Ouyang et al., 2022), ChatGPT (OpenAI, 2022), and GPT4 (OpenAI, 2023). We use a standard and a CoT prompt (Kojima et al., 2022). See implementation details in Appendix B.

6.2 Main Evaluation

We train LMs on each entailment relation separately and present results on ABSPYRAMID$_{[\text{DET}]}$ in Table 4. We observe that fine-tuned LMs can detect abstraction knowledge of *Noun-Entail* with impressive performance. For example, Llama2-Chat (13B) correctly classifies 88.20% of the test data. Meanwhile, models struggle to achieve similar scores on *Verb-Entail* relation. The difficulty of *Verb-Entail* might come from the diversity of word senses we collected from WordNet.

NLI models show some zero-shot ability, especially on *Noun-Entail* and *Event-Entail*. For instance, DeBERTa-large-mnli achieves an accuracy of 73.18% on *Noun-Entail* higher than that of "random" and "majority vote." This finding might be due to some similarity between NLI and our task. Moreover, fine-tuning NLI models cannot improve performance compared with LMs in **PLM + FT**.

Besides, fine-tuned LLMs can obtain scores comparable to or even higher than fully fine-tuned models, whilst we only tuned 0.3-0.5% parameters with LoRA. The performance only improves marginally when we increase the parameters, such as Llama2 (7B) to Llama2 (13B). Meanwhile, the instruction-tuned counterparts cannot lead to distinct increases

Methods	Backbone	Noun Acc	Noun Ma-F1	Noun AUC	Verb Acc	Verb Ma-F1	Verb AUC	Event Acc	Event Ma-F1	Event AUC
Random	-	50.00	49.56	50.00	50.00	49.95	50.00	50.00	48.98	50.00
Majority Vote	-	59.30	-	50.00	53.15	-	50.00	64.14	-	50.00
NLI + Zero	BART-large-mnli	71.24	68.13	75.67	56.25	47.17	62.33	70.69	65.81	69.33
	RoBERTa-large-mnli	68.66	63.18	75.42	55.73	45.54	61.27	70.47	63.07	68.60
	DeBERTa-base-mnli	68.77	65.81	72.79	56.42	48.08	61.55	66.30	62.88	66.40
	DeBERTa-large-mnli	73.18	71.08	78.12	56.93	49.28	63.16	66.82	64.03	68.27
NLI + FT	BART-large-mnli	85.75	85.12	90.80	64.96	64.96	68.60	74.61	69.75	77.71
	RoBERTa-large-mnli	86.15	85.34	90.87	64.61	64.26	69.46	76.88	70.73	77.94
	DeBERTa-base-mnli	85.59	84.61	90.43	65.50	65.47	69.87	76.98	70.12	77.90
	DeBERTa-large-mnli	86.62	85.83	91.00	**66.04**	**65.96**	**70.51**	76.48	69.96	77.42
PLM + FT	BERT-base	85.09	84.14	89.94	64.26	64.20	68.06	76.45	69.94	78.22
	BERT-large	85.94	85.12	90.37	63.58	63.58	68.03	75.27	69.61	77.57
	RoBERTa-base	84.23	83.25	89.58	63.55	63.53	68.12	76.53	70.41	77.62
	RoBERTa-large	85.27	84.44	90.59	64.98	64.98	69.23	77.09	70.56	78.07
	DeBERTa-base	84.09	83.03	89.74	63.50	63.45	68.03	75.75	69.57	77.30
	DeBERTa-large	86.89	86.11	90.98	65.54	65.52	69.11	76.69	70.31	78.06
LLM + LoRA	Falcon (7B)	87.06	86.36	91.42	63.92	63.79	68.06	75.83	70.51	77.77
	Falcon-Ins (7B)	86.04	85.43	91.10	64.00	63.96	68.53	76.50	70.72	77.50
	Mistral (7B)	87.62	87.05	91.53	65.08	64.66	69.58	77.24	70.57	77.97
	Mistral-Ins (7B)	87.59	86.99	91.42	64.81	64.78	69.51	77.22	70.69	78.52
	Llama2 (7B)	87.56	86.82	91.52	65.07	64.79	69.27	76.45	70.53	78.28
	Llama2-Chat (7B)	86.71	86.17	91.79	64.96	64.54	68.95	76.80	70.15	77.92
	Llama2 (13B)	88.03	87.40	**92.31**	65.13	64.64	69.50	76.87	**70.83**	**79.34**
	Llama2-Chat (13B)	**88.20**	**87.49**	92.05	65.07	65.00	69.74	**77.27**	70.82	78.60
LLM API	GPT 4	80.50	78.70	-	56.30	53.84	-	71.30	66.89	-
	GPT 3.5	67.00	62.45	-	56.30	55.90	-	65.60	58.23	-
	ChatGPT	74.00	72.27	-	56.30	55.71	-	68.20	63.22	-
	ChatGPT (CoT)	62.90	62.88	-	56.20	53.89	-	67.30	61.47	-
	ChatGPT (10-shot ICL)	76.10	74.60	-	58.60	58.51	-	68.90	60.51	-
	ChatGPT (CoT + 10-shot)	75.40	74.08	-	59.20	58.91	-	68.20	62.70	-

Table 4: Performance on the test set of ABSPYRAMID$_{[\text{DET}]}$. We trained models on three entailment relations separately. We bold the best score and underline the second-best score. Acc, Ma-F1, and AUC denote Accuracy, Macro F1-score, and ROC-AUC. See the performance on the validation set in Appendix C.1.

but some fluctuations as they learned more about the instruction following and conversations, which are irrelevant to our task.

6.3 Analysis of ChatGPT Series Models

We can see that ChatGPT and GPT3.5 obtain acceptable performance on ABSPYRAMID$_{[\text{DET}]}$ in the zero-shot scenario (Table 4), such as accuracy scores of 74.00% and 67.00% on *Noun-Entail*. However, the ChatGPT series models still lag behind fine-tuned LMs by a large margin, although GPT4 performs better than ChatGPT. Meanwhile, we tested the performance of ChatGPT with ten exemplars under the in-context learning setup, denoted as "ChatGPT (10-shot ICL)." With exemplars, the scores of ChatGPT are raised by 2-3 points but not a substantial improvement since the answer format (i.e., "Yes" or "No") is simple to understand without exemplars.

To explore if the ChatGPT can explain its own decisions, we examine ChatGPT with zero-shot chain-of-thought prompting signified as "ChatGPT (CoT)," where it is asked to explain given words first and then give the answer. Each metric exhibits varying levels of decline, with particular emphasis on *Noun-Entail*. This indicates that ChatGPT cannot explain and provide an answer simultaneously. We conduct an error analysis, as illustrated in Figure 3, to unravel why. The examples show that ChatGPT can explain the meanings of given words but yields hallucinations (Ji et al., 2023; Huang et al., 2023) when concluding. We discover that providing a few exemplars can assist, indicated as "ChatGPT (CoT + 10-shot)" in Table 4. We present all prompts and verify the robustness of zero-shot and CoT prompts in Appendix C.2.

6.4 Multi-Relation Learning

While prior experiments treated each relation separately, we train all entailment relations jointly in

LLM + LoRA	Noun Acc	Noun Ma-F1	Noun AUC	Verb Acc	Verb Ma-F1	Verb AUC	Event Acc	Event Ma-F1	Event AUC	All Acc	All Ma-F1	All AUC
Falcon (7B)	87.11	86.31	91.26	64.68	64.34	69.50	76.55	70.47	78.52	78.15	76.53	84.78
Falcon-Ins (7B)	87.07	86.30	90.91	64.71	64.70	69.16	**77.22**	70.95	78.26	78.28	76.92	84.64
Mistral (7B)	87.77	87.01	91.68	**65.96**	65.60	**70.34**	76.61	70.91	78.88	78.71	77.15	85.40
Mistral-Ins (7B)	87.80	87.09	91.47	65.44	65.35	69.94	77.08	71.08	**79.50**	78.75	77.37	85.38
Llama2 (7B)	87.92	87.09	**91.80**	64.95	64.47	69.59	77.16	71.05	78.75	78.69	76.95	85.39
Llama2-Chat (7B)	87.56	86.79	91.79	64.11	63.98	69.48	76.55	70.53	77.84	78.09	76.98	85.00
Llama2 (13B)	**88.02**	**87.41**	91.73	65.84	**65.84**	70.16	77.11	71.13	78.93	**78.99**	**77.83**	**85.73**
Llama2-Chat (13B)	87.76	87.00	91.59	65.08	64.87	70.02	76.98	**71.16**	79.39	78.67	77.17	85.49

Table 5: The performance of LLMs on the test set of ABSPYRAMID$_{\text{[DET]}}$ under the multi-relation setting. We bold the best score and underline the second-best score. See Appendix C.1 for performance on validation sets.

Models	Acc	Ma-F1	AUC	APS
Aug MC	-	-	-	18.70
CNCE MC	-	-	-	19.50
EGT2	-	-	-	31.90
Falcon (7B)	67.55	63.82	80.06	$39.97_{\uparrow 8.07}$
Mistral (7B)	79.32	**72.66**	**81.42**	$\mathbf{53.25}_{\uparrow \mathbf{21.35}}$
Llama2 (7B)	78.69	71.07	79.51	$44.25_{\uparrow 12.35}$
Llama2 (13B)	**82.11**	71.25	79.84	$45.11_{\uparrow 13.21}$

Table 6: Zero-shot performance on Levy/Holt dataset with LLMs fine-tuned on our dataset. **APS** is average precision score when precision > 0.5 and shows improvements compared with EGT2.

Head Event: The oil futures began trading on exchange.
Abstract Concept: Derivatives

ChatGPT CoT:
✅ **1. Oil futures:** contracts to buy or sell a specified quantity of oil at a future date...
✅ **2. Derivatives:** financial contracts whose value is derived from an underlying asset...
❌ **3. Answer:** No, "derivatives" is not broader.

Head Event: PersonX always slept sitting up.
Abstract Concept: Sleeping style

ChatGPT CoT:
✅ **1. PersonX always slept sitting up:** describes the sleeping habit of PersonX...
✅ **2. Sleeping style:** a way or pattern in which a person sleeps...
❌ **3. Answer:** No.

Figure 3: Error Analysis. We find hallucinations within zero-shot CoT of ChatGPT with correct explanations but wrong conclusions.

this section. The results in Table 5 show that LLMs can learn abstraction knowledge of multiple relations, with performance comparable to that of training on each relation separately (Table 4). Generally, Llama2 (13B) performs best on the merged test set, while varying models get higher performance on each entailment relation. Comparing Llama2 (7B) with Llama2 (13B), we again affirm that scaling up models only leads to marginal improvements.

6.5 Transferring to Other Sources

This section investigates whether the abstraction knowledge from our benchmark can be transferred to other tasks that require the abstraction knowledge (Berant et al., 2011; He et al., 2022).

Verb Entailment Graph: In this task, we evaluate models on the primarily used Levy/Holt dataset (Levy and Dagan, 2016; Holt, 2018), whose statistics are shown in Table 2. We directly experiment with the LLMs fine-tuned on our data (under the multi-relation setting in Section 6.4) to test the zero-shot transferring ability. Following previous works (Hosseini et al., 2021), we also compute the metric "average precision score" when precision is higher than 50%. As shown in Table 6, LLMs fine-tuned on our dataset surpass previous works a lot, including Aug MC (Hosseini et al., 2018), CNCE MC (Hosseini et al., 2019), and EGT2 (Chen et al., 2022). For example, Mistral (7B) achieves the best APS of 53.25, higher than the strongest baseline, EGT2, by over 20 points.

We further test whether knowledge can be transferred in the fine-tuning setup. We continually fine-tune with LoRA LLMs that are first trained on our dataset. They are compared with LLMs fine-tuned from pre-trained configurations. Since the Levy/Holt dataset does not own a training set, we treat the validation set as the training set and do not tune hyperparameters. From Figure 4, the results show that training on our benchmark significantly boosts the performance of LLMs on all metrics. Particularly, the average precision score of Llama2 (7B) rises from 61.0 to 75.8 if we first fine-tune it on our benchmark. These experiments demonstrate that our benchmark is comprehensive to boost performance in both zero-shot and fine-tuning setups.

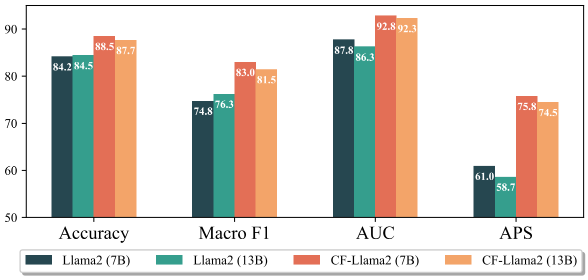


Figure 4: The fine-tuning performance on the Levy/Holt dataset. **CF** stands for continually fine-tuning.

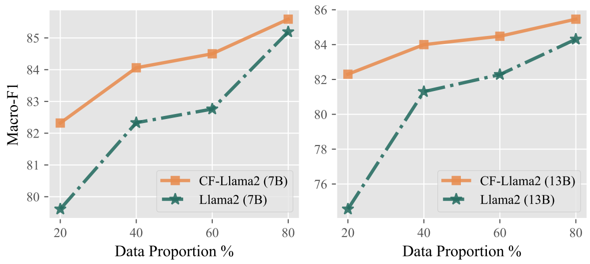


Figure 5: Few-shot performance on AbstractATOMIC. **CF** stands for continually fine-tuning.

AbstractATOMIC To further verify the comprehensiveness of our benchmark, we fine-tuned LLMs under the few-shot setting on the AbstractATOMIC dataset, where we start from 20% of training data and increase the proportion by 20% each time. Similarly, we fine-tuned two categories of LLMs: pre-trained models and models initially trained on our dataset. While only a modest fraction of our dataset falls under the social domain (in Table 2), we discover that our dataset still can significantly enhance performance on AbstractATOMIC, as displayed in Figure 5. The results show that our dataset contains comprehensive abstract knowledge, which can help models generalize to a specific domain. We include full results of more LLMs on both Levy/Holt and AbstractATOMIC datasets in Appendix C.3.

7 Abstraction Generation Experiment

In this section, we evaluate representative LMs on the ABSPYRAMID$_{[\text{GEN}]}$.

7.1 Experiment Setup

Evaluation Metric BLEU-1, BLEU-2 (Papineni et al., 2002), ROUGE-2, ROUGE-L (Lin, 2004), and Meteor (Banerjee and Lavie, 2005) are computed to automatically evaluate all models.

Language Models We evaluated representative LMs, including GPT-J (6B) (Wang and Komatsuzaki, 2021), Falcon (7B) and Falcon-Instruct (7B) (Penedo et al., 2023), Llama2 (7B, 13B) and Llama2-Chat (7B, 13B) (Touvron et al., 2023), GPT2, and GPT2-medium/large/XL (Radford et al., 2019). See implementation details in Appendix B.

Models	B-1	B-2	R-2	R-L	Meteor
GPT2	27.42	10.56	4.34	25.03	21.72
GPT2-medium	33.86	15.52	6.64	31.37	25.30
GPT2-large	49.23	29.64	16.80	48.36	35.44
GPT2-XL	53.90	32.39	**18.54**	53.73	<u>38.45</u>
GPT-J (6B)	55.65	31.19	15.20	54.42	36.70
Falcon (7B)	54.63	30.64	14.46	54.15	36.36
Falcon-Ins (7B)	53.18	30.15	14.96	51.90	35.17
Llama2 (7B)	56.56	33.03	16.48	56.37	37.67
Llama2-Chat (7B)	57.11	34.42	16.31	54.87	37.34
Llama2 (13B)	**58.73**	**36.28**	<u>17.63</u>	**57.45**	**39.47**
Llama2-Chat (13B)	<u>58.46</u>	<u>34.54</u>	16.39	<u>56.47</u>	37.95

Table 7: Results on the test set of ABSPYRAMID$_{[\text{GEN}]}$. B-1/2, R-2/L denote BLEU-1/2, ROUGE-2/L.

7.2 Main Evaluation

We present the overall performance of all language models in Table 7. We ascertain that fine-tuned language models can perform fairly well on our generation dataset. For example, Llama2 (13B) achieves the best BLEU-2 score, where 36.28% of generated bi-grams are covered by the references. Unlike abstraction detection, increasing the number of parameters exerts a more significant effect on abstraction generation. For example, GPT2-XL (1.56B) gets the highest ROUGE-2 score, which is times higher than GPT2 (117M) and GPT2-medium (345M). Also, the performance of Llama2 (13B) is 1-3 points higher on all metrics than Llama2 (7B). Another noteworthy point is that instruction tuning does not help abstraction generation, exemplified by Llama2 (13B) getting higher metrics scores than Llama2-Chat (13B). We also include the performance on data of each entailment relation in Appendix C.4. Similar to abstraction detection, we can find that models perform better on *Noun-Entail* than other relations.

8 Conclusion

In this paper, we introduce ABSPYRAMID to evaluate LLMs' abstraction ability. A scalable pipeline is designed to curate abstraction knowledge for three components of events. We carry out extensive experiments to demonstrate the comprehensiveness of our benchmark and provide valuable insights into the abstraction abilities of LLMs.

Limitations

Our AbsPyramid incorporates extensive abstraction knowledge of events from ASER for nouns, verbs, and events. An open question is how to interleave the abstraction knowledge into the eventuality knowledge represented as explicit discourse relations in ASER. For the same event, we can have different levels of abstraction depending on the current context provided by eventuality knowledge. In the event "I drink milk," "milk" can be abstracted as "beverage" under the situation that "I am thirsty." In contrast, "milk" is better to be considered a kind of "dairy product" if "I want to get more nutrition." Other knowledge can also be considered, such as factual knowledge (Sun et al., 2023) and commonsense knowledge (Sap et al., 2019; Hwang et al., 2021; West et al., 2022).

Representative LLMs are evaluated in our experiments. We leave for future work about building models with stronger abstraction abilities, including some sophisticated prompting methods (Yao et al., 2023; Long, 2023; Besta et al., 2023), combining LLMs with smaller LMs (Xu et al., 2023), semi-supervised learning (Wang et al., 2023), retrieval augmented generation (Lewis et al., 2020b).

Ethics Statement

When constructing AbsPyramid, we sample head events from ASER (Zhang et al., 2020, 2022), an open-sourced eventuality graph. We only sampled eventualities extracted from Wikipedia and NYT, which are open-access. We carried out human annotation on Amazon Mechanical Turk (MTurk). Our payment rate is 1.2 USD for each HIT, which fulfills the minimum wage requirement and shows that annotators are fairly paid.

References

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, Ann Arbor, Michigan. Association for Computational Linguistics.

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.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

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

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with python.

Samuel R. 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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023. PropSegmEnt: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8874–8893, Toronto, Canada. Association for Computational Linguistics.

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

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.

Eliana Colung and Linda B Smith. 2003. The emergence of abstract ideas: Evidence from networks and babies. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1435):1205–1214.

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, Brussels, Belgium. Association for Computational Linguistics.

R Cooper, R Crouch, J van Eijck, C Fox, J van Genabith, J Jaspars, H Kamp, M Pinkal, D Milward, M Poesio, et al. 1996. Fracas: A framework for computational semantics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7358–7370. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Yu Gong, Kaiqi Zhao, and Kenny Zhu. 2016. Representing verbs as argument concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

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.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

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.

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*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing 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, Hong Kong, China. Association for Computational Linguistics.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and

Yejin Choi. 2021. (comet-) atomic 2020: on symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

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*.

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

Daniel Z. Korman, Eric Mack, Jacob Jett, and Allen H. Renear. 2018. Defining textual entailment. *J. Assoc. Inf. Sci. Technol.*, 69(6):763–772.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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

Tianyi Li, Sabine Weber, Mohammad Javad Hosseini, Liane Guillou, and Mark Steedman. 2022. Cross-lingual inference with A chinese entailment graph. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1214–1233. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jieyi Long. 2023. Large language model guided tree-of-thought.

Bill MacCartney et al. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June 28-29, 2007*, pages 193–200. Association for Computational Linguistics.

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*.

Adam Meyers. Annotation guidelines for nombank–noun argument structure for propbank 2007.

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.

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

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report.

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.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

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*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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*, pages 11–47.

Evan Sandhaus. 2008. The new york times annotated corpus.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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.

Uli Sauerland and Penka Stateva, editors. 2007. *Presupposition and Implicature in Compositional Semantics*. Palgrave-Macmillan, Houndmills, Basingstoke, Hampshire.

Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the twenty-second international joint conference on artificial intelligence-volume volume three*, pages 2330–2336.

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*.

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*.

Benjamin Van Durme, Phillip Michalak, and Lenhart Schubert. 2009. Deriving generalized knowledge from corpora using wordnet abstraction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 808–816.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Weiqi Wang, Tianqing Fang, Baixuan Xu, Chun Yi Louis Bo, Yangqiu Song, and Lei Chen. 2023. CAT: A contextualized conceptualization and instantiation framework for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13111–13140, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

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.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2023. Small models are valuable plug-ins for large language models. *arXiv preprint arXiv:2305.08848*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

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.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

A Data Curation Details

A.1 ASER Cleaning

Since ASER is an eventuality graph automatically extracted from diverse corpora, some noisy extraction results exist. Thus, we design a few rules to clean some frequent noise categories in ASER.

First, we found that many eventualities are noisy due to incompleteness. For example, “the norman army weakened,” an eventuality extracted from Wikipedia, misses the linking verb “was” in the passive voice. To solve this, we re-parse each eventuality and remove eventualities whose dependency graph changes in the re-parsing stage. With this rule, we remove a lot of incomplete eventualities.

Then, we design four lexical rules for noisy eventualities: (1) We find that many eventualities with the `s-v` pattern (see (Zhang et al., 2022) for definition) contain light verbs. We remove those eventualities since they lack semantic meanings, such as “they do.” (2) We find that the parsing algorithm of ASER can extract eventualities from subordinate clauses but cannot link relatives to antecedents. For example, “who won the competition” is extracted from the sentence “Bob is a painter who won the competition” without replacing “who” with “Bob.” We remove all eventualities starting with relatives. (3) ASER also contains some eventualities that are totally composed of stopwords. We remove them since they also do not have too many semantic meanings, such as “She just won.” (4) We remove eventualities containing URLs and HTML tags.

In detail, the light verbs we use are `do`, `give`, `have`, `make`, `get`, and `take`, as well as their inflections, such as `doing` and `has`. The relatives we use are `how`, `what`, `when`, `where`, `which`, `who`, `why`, `whatever`, `whose`, `whom`, and `if`. The stopword list is accessed by NLTK (Bird et al., 2009).

A.2 Matching Nouns and Verbs

In our benchmark, the abstraction knowledge of *Noun-Entail* and *Verb-Entail* involves identifying nouns and verbs from events. In ASER, each word in the syntactic pattern is classified into word types according to their POS tags, including `noun`, `verb`, `be`, and `preposition`. We use those word types to identify the nouns and verbs. For example, the pattern `subject-verb-object` has word types `noun`, `verb`, and `noun` for each word. Also, we identify modifiers to complete each noun by collecting all words dependent on the noun in the dependency parsing graph, such as “fluffy” in “fluffy cat.”

Task Instruction: In this task, you need to list the hypernyms of an instance. Hypernyms are words that represent broader categories or concepts.
Exemplar Input: 1. Given the sentence "the clinic had resumed its work," what is the list of hypernyms of "clinic?" **Exemplar Output:** (1) medical facility, (2) healthcare center, . . . , (10) diagnostic center.
Following Exemplars: Exemplar 2, Exemplar 3, . . . , Exemplar 10
Testing Input: 11. Given the sentence **[HEAD]**, what is the list of hypernyms of **[INSTANCE]**?

(a) *Noun-Entail*

Task Instruction: In this task, you need to list some abstract descriptions of an event.
Exemplar Input: 1. Which abstract descriptions can the event "PersonX surfs the web" be summarized as? **Exemplar Output:** (1) surfing, (2) surfing the internet, . . . , (10) browsing the internet.
Following Exemplars: Exemplar 2, Exemplar 3, ..., Exemplar 10
Testing Input: 11. Which abstract descriptions can the event **[HEAD]** be summarized as?

(b) *Event-Entail*

Table 8: The prompt we used to collect abstract concepts from ChatGPT for *Noun-Entail* and *Event-Entail* relations. Two placeholders **[HEAD]** and **[ISNTANCE]** will be replaced with real head events and instances. We present the prompt in the dialogue format. Please concatenate all utterances to form the prompt of GPT3.5.

Task Instruction: In this task, you need to list some related nouns but not hypernyms. Hypernyms are words that represent broader categories or concepts.
Exemplar Input: 1. Given the sentence "the clinic had resumed its work," please list related nouns of "clinic" but not hypernyms. **Exemplar Output:** (1) patients, (2) doctors, . . . , (10) mask.
Following Exemplars: Exemplar 2, Exemplar 3, ..., Exmplar 10
Testing Input: 11. Given the sentence **[HEAD]**, please list related nouns of **[INSTANCE]** but not hypernyms.

(a) *Noun-Entail*

Task Instruction: In this task, you need to list some related phrases but not abstract descriptions of an event.
Exemplar Input: 1. Please list related phrases of the event "PersonX surfs the web" but not abstract descriptions of it. **Exemplar Output:** (1) typing a URL, (2) website, . . . , (10) bandwidth.
Following Exemplars: Exemplar 2, Exemplar 3, ..., Exemplar 10
Testing Input: 11. Please list related phrases of the event **[HEAD]** but not abstract descriptions of it.

(b) *Event-Entail*

Table 9: The prompt we used to collect challenging negative examples from ChatGPT for *Noun-Entail* and *Event-Entail* relations.

We also take care of some special cases where eventualities contain some transparent nouns (Meyers), such as "I have **a lot of food**." In this case, we identify "food" as an instance instead of "lot." Verbs also have similar constructions, such as "I **am going to sleep**." In this example, we identify "sleep" as an instance instead of "going."

A.3 Prompts for Collecting Data

We provide the prompt template used in collecting abstract concepts in Table 8 and the prompt template used in collecting negative examples in Table 9.

A.4 Annotation Details

There are two qualification tests to choose workers to maintain rigorous quality control. First, we invited annotators who meet the following conditions to take our qualification examinations: 1) an approval rate of above 95% and 2) at least a thousand approved HITs. In the second round, qualification questions, including effortless and tricky examples, are collected by this paper's authors, who clearly understand abstract tuples. The experts annotate 200 tuples for each relation. An annotator should correctly answer 18 of 20 questions to pass the second round test.

In our main annotation, we assign each tuple to 5 annotators in the first round of annotations. We manually inspect their annotation quality and disqualify those annotators who cannot continue to annotate with high accuracy. The annotations from those disqualified annotators are then discarded for quality control. For higher quality, we also introduce two rounds of refinement. We reannotate the discarded votes in the first round of refinement. In the second round, we request annotators to reannotate the tuples that do not reach an agreement (i.e., 2 or 3 out of 5 annotators vote for valid). After this, we discard examples that annotators still do not agree on. We show the full text of instructions provided to annotators in Figure 6.

During our massive annotation process, 5153 annotators participated in qualification tests, with 551 (10.7%) annotators passing them. The IAA score of pairwise agreement proportion is 77.62%,

LLMs	Noun		Verb		Event	
	Acc	Ma-F1	Acc	Ma-F1	Acc	Ma-F1
GPT 4	62.70	62.47	57.70	57.54	66.20	64.06
GPT 3.5	66.10	62.72	54.10	53.94	67.40	59.57
ChatGPT	67.40	66.04	55.20	55.04	67.60	63.36
+ CoT	56.70	56.67	54.00	52.39	61.30	60.13

Table 10: Results of *NLI* prompt on ABSPYRAMID$_{\text{[DET]}}$. We mark scores higher than scores of *Abs.* prompt in Table 4 with red color. We can see that most scores are inferior.

***Noun-Entail*, *Verb-Entail* and *Event-Entail*:** Identify entailment and provide a "Yes" or "No" response. Entailment is about determining whether a "hypothesis" is true given a "premise." Given the premise **[HEAD]**, can we know the hypothesis **[TAIL]**?

(a) Zero-Shot Prompt

***Noun-Entail*, *Verb-Entail* and *Event-Entail*:** Identify entailment, which is about determining whether a "hypothesis" is true given a "premise." Given the premise **[HEAD]**, can we know the hypothesis **[TAIL]**? Step 1: Let's think about meanings of those sentences. Step 2: Provide a "Yes" or "No" response.

(b) CoT Prompt

Table 11: The NLI-format prompt. Results of this prompt is shown in Table 10. Placeholders **[HEAD]** and **[TAIL]** will be replaced with real head events and tail events.

***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 **[CONCEPT]** encompass **[INSTANCE]**?

***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 **[CONCEPT]** encompass **[INSTANCE]**?

***Event-Entail*:** Identify abstract descriptions of specific sentences, and provide a "Yes" or "No" response. Can we consider **[CONCEPT]** as an abstract description of the sentence **[HEAD]**?

(a) Zero-Shot Prompt

***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 **[CONCEPT]** encompass **[INSTANCE]**? Step 1: Let's think about the meanings of those words. Step 2: Provide a "Yes" or "No" response.

***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 **[CONCEPT]** encompass **[INSTANCE]**? Step 1: Let's think about the meanings of those words. Step 2: Provide a "Yes" or "No" response.

***Event-Entail*:** Identify abstract descriptions of specific sentences. Can we consider **[CONCEPT]** as an abstract description of the sentence **[HEAD]**? Step 1: Let's think about the meanings of the sentence and the abstract description. Step 2: Provide a "Yes" or "No" response.

(b) CoT Prompt

Table 12: The default prompt we used (i.e., *Abs.* prompt) to test GPT3.5, ChatGPT, and GPT4. The results of this prompt are shown in Table 4. Placeholders **[HEAD]**, **[INSTANCE]**, and **[CONCEPT]** will be replaced with real head events, instances, and abstract concepts.

and Fleiss's κ is 0.54.

B Implementation Details

First, we discuss details shared in both abstraction detection and abstraction generation experiments. We access open-source language models using Transformers (Wolf et al., 2020) and fine-tune them on 8 NVIDIA A100 (80G) GPUs. LLMs with 7B and 13B parameters are loaded with BF16. The best checkpoint is selected according to the sum of all metrics on the validation set. When fine-tuning LLMs with LoRA, we only add new parameters to attention layers with the rank and α equal to 64 and 128. We grid search the learning rate of 5e-6, 1e-5, 5e-5, and batch sizes of 64 and 128.

Here are some details specific to abstraction detection experiments. When fine-tuning **NLI models**, we re-use the classification layer with "Entailment" and "Neutral" for valid and invalid, respectively. We access ChatGPT, GPT4, and GPT3.5 via OpenAI API[4], with specific versions being `gpt-3.5-turbo-0613`, `gpt-4-0613`, and `gpt-3.5-turbo-instruct-0914`. They are evaluated on one thousand examples that we randomly sampled from the testing set of each relation due to the trade-off between API expenses and our evaluation's precision. In addition, we provide ChatGPT with ten exemplars for in-context learning.

C Experimental Results

In this appendix, we collect supplementary abstraction detection and generation results.

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

Models	Acc	Ma-F1	AUC	APS
Falcon (7B)	82.93	74.57	86.55	57.46
Mistral (7B)	84.56	76.67	88.60	62.78
Llama2 (7B)	84.20	74.81	87.75	60.98
Llama2 (13B)	84.47	76.28	86.27	58.69
CF-Falcon (7B)	87.19	80.52	91.21	71.21
CF-Mistral (7B)	88.28	82.14	92.64	77.78
CF-Llama2 (7B)	88.55	83.04	92.83	75.83
CF-Llama2 (13B)	87.70	81.48	92.33	74.51

Table 13: The fine-tuning performance of LLMs on the Levy/Holt dataset. **CF** stands for continually fine-tuning.

C.1 Validation Results on Abstraction Detection

We collect the performance of LMs trained on each entailment relation separately on the validation set of the AbsPyramid$_{\text{[DET]}}$ in Table 20. Then, we present the performance of LMs trained on merged data of all entailment relations on the validation set in Table 19.

C.2 ChatGPT Prompt Robustness

First, we ask GPT3.5, ChatGPT, and GPT4 whether an abstract concept is valid as the default prompt (denoted as *Abs.* prompt). The prompt is presented in Table 12, and its results are shown in Table 4. Meanwhile, we design another prompt in NLI format, treating the head and tail events as the premise and hypothesis (denoted as *NLI* prompt). This prompt is presented in Table 11. As shown in Table 10, the performance of the *NLI* prompt is inferior to the *Abs.* prompt on most metrics, showing the robustness of the *Abs.* prompt.

C.3 Full Results of Transferring to Other Sources

Here, the full fine-tuning performance of all LLMs on the Levy/Holt dataset is shown in Table 13. Also, we provide the full results of all pre-trained LLMs on AbstractATOMIC in Table 14 and results of LLMs that initially fine-tuned on our dataset in Table 15.

Models	Shot	Acc	Ma-F1	AUC
Falcon (7B)	0%	59.39	41.01	61.18
	20%	73.41	72.36	80.20
	40%	81.17	80.36	88.73
	60%	82.37	81.76	89.73
	80%	83.13	82.71	91.20
Mistral (7B)	0%	41.88	31.44	53.71
	20%	83.14	82.64	90.56
	40%	84.12	83.90	92.57
	60%	85.66	85.30	92.98
	80%	85.72	85.42	93.66
Llama2 (7B)	0%	59.39	41.01	61.18
	20%	80.28	79.61	87.89
	40%	82.93	82.33	90.96
	60%	83.12	82.76	91.41
	80%	85.67	85.19	92.97
Llama2 (13B)	0%	55.94	38.81	43.41
	20%	75.59	74.56	82.19
	40%	81.87	81.30	89.71
	60%	82.98	82.28	90.44
	80%	84.93	84.31	92.39

Table 14: The few-shot performance on the test set of AbstractATOMIC dataset. LLMs are loaded from pre-trained configurations.

Models	Shot	Acc	Ma-F1	AUC
Falcon (7B)	0%	64.22	64.22	72.80
	20%	81.11	80.54	89.01
	40%	83.49	82.98	91.11
	60%	83.95	83.45	91.66
	80%	84.67	84.22	92.24
Mistral (7B)	0%	64.81	64.78	73.60
	20%	84.43	84.03	91.73
	40%	85.85	85.40	92.88
	60%	86.24	85.75	93.23
	80%	86.61	86.20	93.71
Llama2 (7B)	0%	62.40	62.13	71.65
	20%	82.70	82.32	90.43
	40%	84.51	84.06	91.90
	60%	84.91	84.50	92.26
	80%	85.97	85.59	93.13
Llama2 (13B)	0%	64.28	64.25	71.35
	20%	82.76	82.30	90.23
	40%	84.50	84.00	91.88
	60%	84.91	84.48	92.22
	80%	85.87	85.46	93.01

Table 15: The few-shot performance on the test set of AbstractATOMIC dataset. LLMs are initially trained on AbsPyramid$_{\text{[DET]}}$.

C.4 Generation Results by Relation

To carry out a more thorough evaluation of LMs' ability to generate abstraction knowledge, we also provide performance by entailment relations *Noun-Entail*, *Verb-Entail*, and *Event-Entail* in Tables 16 to 18, respectively.

Models	**B-1**	**B-2**	**R-2**	**R-L**	**Meteor**
GPT2	33.67	11.63	3.35	30.75	20.04
GPT2-medium	39.15	15.64	6.09	39.43	24.82
GPT2-large	55.79	30.16	15.18	57.31	37.93
GPT2-XL	62.47	33.94	18.70	64.67	42.30
GPT-J (6B)	67.47	35.65	15.47	67.17	41.32
Falcon (7B)	68.67	36.48	16.25	71.62	43.63
Falcon-Ins (7B)	63.92	32.08	13.51	65.31	39.49
Llama2 (7B)	65.80	33.73	17.28	70.29	43.47
Llama2-Chat (7B)	70.07	39.08	18.12	71.51	45.00
Llama2 (13B)	68.81	34.91	18.02	71.04	45.17
Llama2-Chat (13B)	68.71	33.60	16.67	70.54	43.79

Table 16: Generation results on data of *Noun-Entail* in the test set of ABSPYRAMID$_{[\text{GEN}]}$. B-1/2, R-2/L denote BLEU-1/2, ROUGE-2/L, respectively.

Models	**B-1**	**B-2**	**R-2**	**R-L**	**Meteor**
GPT2	5.44	0.00	0.00	5.79	18.21
GPT2-medium	11.46	1.25	0.18	11.77	21.00
GPT2-large	40.34	44.37	12.23	36.98	30.58
GPT2-XL	44.14	39.47	10.77	42.62	31.99
GPT-J (6B)	40.82	31.46	5.11	40.33	27.66
Falcon (7B)	36.88	28.77	3.83	37.01	26.06
Falcon-Ins (7B)	38.49	38.38	6.93	36.68	26.30
Llama2 (7B)	43.92	36.47	5.29	41.94	27.45
Llama2-Chat (7B)	36.68	26.58	3.83	36.79	24.32
Llama2 (13B)	45.18	43.53	6.75	43.90	29.85
Llama2-Chat (13B)	42.25	35.16	5.84	41.94	27.76

Table 17: Generation results on data of *Verb-Entail* in the test set of ABSPYRAMID$_{[\text{GEN}]}$. B-1/2, R-2/L denote BLEU-1/2, ROUGE-2/L, respectively.

Models	**B-1**	**B-2**	**R-2**	**R-L**	**Meteor**
GPT2	35.24	10.93	10.86	42.19	28.06
GPT2-medium	44.12	17.54	15.28	46.23	31.19
GPT2-large	50.39	25.52	24.38	52.01	38.57
GPT2-XL	53.92	29.73	27.98	54.69	41.96
GPT-J (6B)	56.28	29.24	27.38	56.96	42.51
Falcon (7B)	55.15	28.24	25.53	54.96	40.63
Falcon-Ins (7B)	54.90	27.88	26.63	55.10	41.10
Llama2 (7B)	57.48	32.16	29.40	58.00	43.56
Llama2-Chat (7B)	60.18	33.52	29.66	57.84	44.51
Llama2 (13B)	59.34	35.82	30.66	58.36	44.74
Llama2-Chat (13B)	61.06	34.88	29.13	58.04	43.74

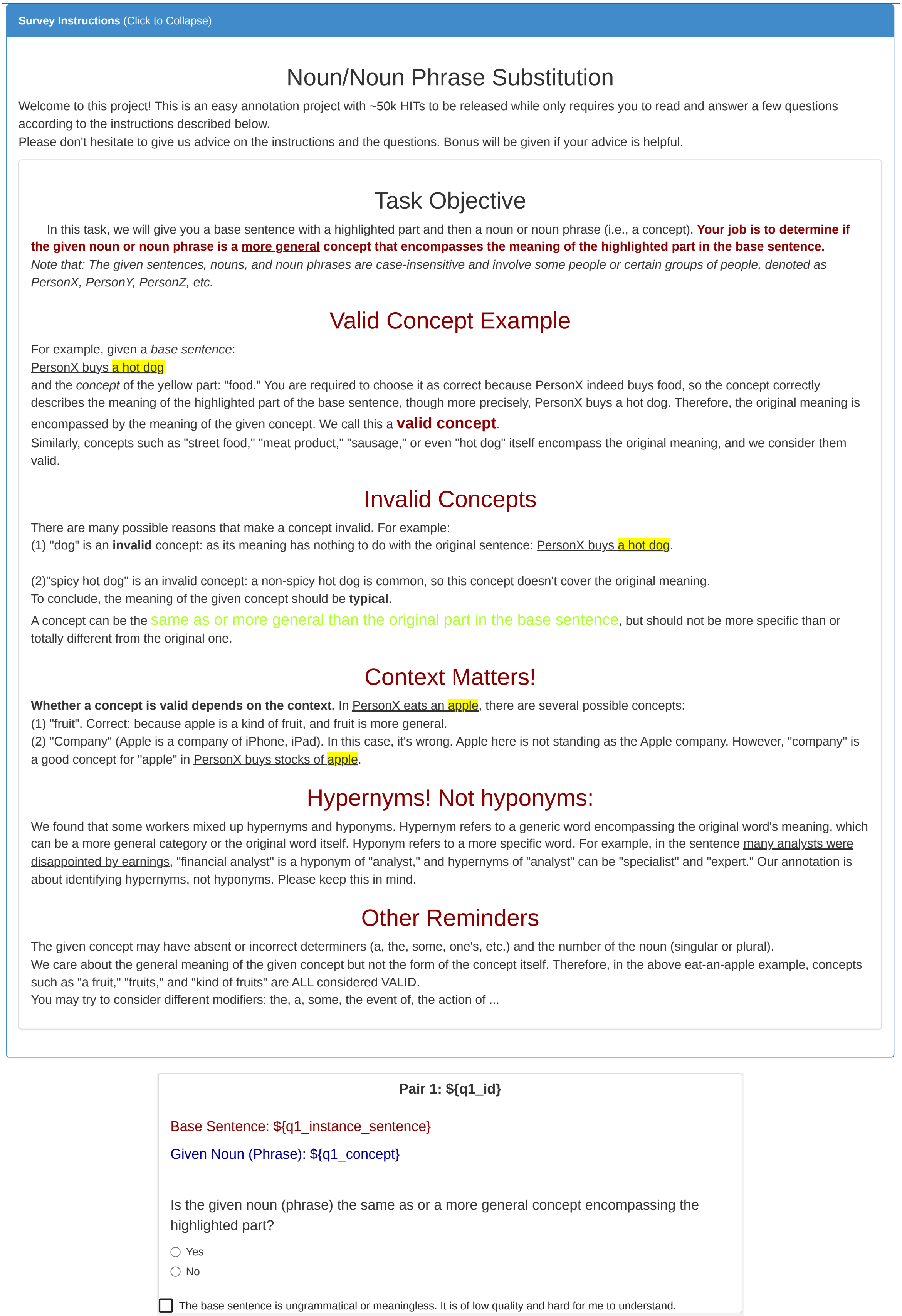
Table 18: Generation results on data of *Event-Entail* in the test set of ABSPYRAMID$_{[\text{GEN}]}$. B-1/2, R-2/L denote BLEU-1/2, ROUGE-2/L, respectively.

LLM + LoRA	Noun			Verb			Event			All		
	Acc	Ma-F1	AUC	Acc	Ma-F1	AUC	Acc	Ma-F1	AUC	Acc	Ma-F1	AUC
Falcon (7B)	88.12	87.55	92.60	64.42	64.15	68.92	77.54	71.84	80.38	78.76	77.38	85.95
Falcon-Ins (7B)	87.62	87.09	92.44	64.61	64.59	69.23	77.39	71.44	80.29	78.52	77.37	85.88
Mistral (7B)	88.90	88.38	92.86	64.61	64.30	69.75	77.95	72.56	81.07	79.28	77.96	86.73
Mistral-Ins (7B)	88.57	88.09	92.77	64.49	64.40	68.76	77.78	72.10	81.02	79.04	77.86	86.50
Llama2 (7B)	88.85	88.29	92.97	64.17	63.84	68.95	77.97	71.95	80.97	79.15	77.71	86.59
Llama2-Chat (7B)	88.37	87.82	92.86	64.07	63.94	68.93	77.39	71.53	79.68	78.78	77.82	86.04
Llama2 (13B)	88.26	87.83	92.85	65.20	65.20	69.48	77.65	71.95	80.57	79.06	78.08	86.57
Llama2-Chat (13B)	88.62	88.09	92.77	65.47	65.31	69.71	77.65	72.11	81.31	79.25	78.01	86.60

Table 19: The performance of LLMs on the validation set of ABSPYRAMID$_{[\mathrm{DET}]}$ under the multi-relation setting.

Methods	Backbone	Noun			Verb			Event		
		Acc	Ma-F1	AUC	Acc	Ma-F1	AUC	Acc	Ma-F1	AUC
Random	-	50.00	49.67	50.00	50.00	49.97	50.00	50.00	49.01	50.00
Majority Vote	-	58.11	-	50.00	52.40	-	50.00	63.94	-	50.00
NLI + Zero	BART-large-mnli	70.44	67.65	75.47	54.84	45.89	62.54	71.32	66.65	71.06
	RoBERTa-large-mnli	67.76	62.61	74.70	54.10	43.55	61.51	70.40	62.65	70.62
	DeBERTa-base-mnli	67.77	65.05	72.35	54.72	46.35	61.34	66.14	62.52	67.21
	DeBERTa-large-mnli	72.85	70.95	78.23	55.68	48.23	62.34	68.35	65.30	70.55
NLI + FT	BART-large-mnli	86.47	86.03	91.92	64.47	64.47	68.53	75.58	71.02	79.63
	RoBERTa-large-mnli	86.93	86.35	91.92	65.16	64.83	69.06	77.75	71.42	80.25
	DeBERTa-base-mnli	86.17	85.42	91.24	64.64	64.61	68.96	77.36	70.66	79.50
	DeBERTa-large-mnli	86.92	86.30	91.78	64.15	64.08	69.30	77.47	71.07	79.65
PLM + FT	BERT-base	85.47	84.78	91.02	63.38	63.32	68.35	77.33	71.06	80.27
	BERT-large	86.65	86.03	91.37	62.96	62.95	67.02	76.16	70.84	79.73
	RoBERTa-base	85.01	84.31	90.76	62.62	62.61	67.04	77.25	71.37	79.75
	RoBERTa-large	86.35	85.80	91.29	62.91	62.91	67.64	77.86	71.53	79.89
	DeBERTa-base	85.22	84.51	90.31	62.28	61.89	67.34	76.85	71.25	79.55
	DeBERTa-large	87.77	87.23	91.91	64.79	64.79	68.49	77.75	71.58	80.05
LLM + LoRA	Falcon (7B)	87.49	86.97	92.33	63.56	63.43	68.13	76.45	71.49	79.50
	Falcon-Ins (7B)	86.57	86.11	92.07	64.15	64.09	68.46	76.17	70.53	78.89
	Mistral (7B)	88.50	88.08	92.63	63.29	62.90	68.16	77.91	71.52	80.58
	Mistral-Ins (7B)	88.31	87.90	92.60	63.71	63.65	68.77	77.91	72.00	80.72
	Llama2 (7B)	88.57	88.06	92.84	63.71	63.32	68.75	76.91	71.36	80.18
	Llama2-Chat (7B)	87.87	87.48	92.92	63.53	63.09	67.79	77.91	71.58	79.79
	Llama2 (13B)	88.64	88.16	93.09	64.08	63.57	69.03	77.43	71.68	80.61
	Llama2-Chat (13B)	88.59	88.03	92.89	64.32	64.23	68.89	77.89	71.62	80.70

Table 20: Performance on the validation set of our ABSPYRAMID$_{[\mathrm{DET}]}$. We trained models on the three entailment relations separately.



Survey Instructions (Click to Collapse)

Noun/Noun Phrase Substitution

Welcome to this project! This is an easy annotation project with ~50k HITs to be released while only requires you to read and answer a few questions according to the instructions described below.
Please don't hesitate to give us advice on the instructions and the questions. Bonus will be given if your advice is helpful.

Task Objective

In this task, we will give you a base sentence with a highlighted part and then a noun or noun phrase (i.e., a concept). **Your job is to determine if the given noun or noun phrase is a more general concept that encompasses the meaning of the highlighted part in the base sentence.**
Note that: The given sentences, nouns, and noun phrases are case-insensitive and involve some people or certain groups of people, denoted as PersonX, PersonY, PersonZ, etc.

Valid Concept Example

For example, given a *base sentence*:
PersonX buys a hot dog
and the *concept* of the yellow part: "food." You are required to choose it as correct because PersonX indeed buys food, so the concept correctly describes the meaning of the highlighted part of the base sentence, though more precisely, PersonX buys a hot dog. Therefore, the original meaning is encompassed by the meaning of the given concept. We call this a **valid concept**.
Similarly, concepts such as "street food," "meat product," "sausage," or even "hot dog" itself encompass the original meaning, and we consider them valid.

Invalid Concepts

There are many possible reasons that make a concept invalid. For example:
(1) "dog" is an **invalid** concept: as its meaning has nothing to do with the original sentence: PersonX buys a hot dog.

(2)"spicy hot dog" is an invalid concept: a non-spicy hot dog is common, so this concept doesn't cover the original meaning.
To conclude, the meaning of the given concept should be **typical**.
A concept can be the same as or more general than the original part in the base sentence, but should not be more specific than or totally different from the original one.

Context Matters!

Whether a concept is valid depends on the context. In PersonX eats an apple, there are several possible concepts:
(1) "fruit". Correct: because apple is a kind of fruit, and fruit is more general.
(2) "Company" (Apple is a company of iPhone, iPad). In this case, it's wrong. Apple here is not standing as the Apple company. However, "company" is a good concept for "apple" in PersonX buys stocks of apple.

Hypernyms! Not hyponyms:

We found that some workers mixed up hypernyms and hyponyms. Hypernym refers to a generic word encompassing the original word's meaning, which can be a more general category or the original word itself. Hyponym refers to a more specific word. For example, in the sentence many analysts were disappointed by earnings, "financial analyst" is a hyponym of "analyst," and hypernyms of "analyst" can be "specialist" and "expert." Our annotation is about identifying hypernyms, not hyponyms. Please keep this in mind.

Other Reminders

The given concept may have absent or incorrect determiners (a, the, some, one's, etc.) and the number of the noun (singular or plural).
We care about the general meaning of the given concept but not the form of the concept itself. Therefore, in the above eat-an-apple example, concepts such as "a fruit," "fruits," and "kind of fruits" are ALL considered VALID.
You may try to consider different modifiers: the, a, some, the event of, the action of ...

Pair 1: ${q1_id}

Base Sentence: ${q1_instance_sentence}

Given Noun (Phrase): ${q1_concept}

Is the given noun (phrase) the same as or a more general concept encompassing the highlighted part?

- ○ Yes
- ○ No

☐ The base sentence is ungrammatical or meaningless. It is of low quality and hard for me to understand.

Figure 6: The full text of instructions provided to annotators on Amazon Mechanical Turk (MTurk). There are ten questions in a Human Intelligence Task (HIT), and we only display one here for brevity.