EVENTS REALM: Event Reasoning of Entity States via Language Models

Evangelia Spiliopoulou[*†] Amazon AWS, AI Labs spilieva@amazon.com

Artidoro Pagnoni[*] Univ. of Washington artidoro@cs.washington.edu

Yonatan Bisk Carnegie Mellon University ybisk@cs.cmu.edu

Eduard Hovy Carnegie Mellon University hovy@cs.cmu.edu

Abstract

This paper investigates models of event implications. Specifically, how well models predict entity state-changes, by targeting their understanding of physical attributes. Nominally, Large Language models (LLM) have been exposed to procedural knowledge about how objects interact, yet our benchmarking shows they fail to *reason* about the world. Conversely, we also demonstrate that existing approaches often misrepresent the surprising abilities of LLMs via improper task encodings and that proper model prompting can dramatically improve performance of reported baseline results across multiple tasks. In particular, our results indicate that our prompting technique is especially useful for unseen attributes (out-of-domain) or when only limited data is available.[1]

1 Introduction

Modeling the effect of actions on entities (*event implications*) is a fundamental problem in AI spanning computer vision, cognitive science and natural language understanding. Most commonly referred to as the Frame Problem (McCarthy and Hayes, 1981), early solutions relied on a set of handcrafted rules and logical statements to model event implications. However, such methods require substantial manual effort and fail to generalize. More recently, modeling event implications has reemerged under the guise of common sense reasoning within NLP (Sap et al., 2019b; Bisk et al., 2020b; Talmor et al., 2019) and action anticipation in Computer Vision (Damen et al., 2018; Bakhtin et al., 2019).

Predicting event implications is a particularly difficult problem due to the complex nature of language and implicit knowledge required to answer such questions. For example, if we are given the sentence *the mug fell on the floor* and we want to determine whether *the mug is broken*, we need to know of several facts such as the material of the mug, the fragility of ceramics, the hardness of the floor, etc. and also how to combine these facts together to **reason** whether the mug will break or not. None of this knowledge is explicitly stated, instead being classified as *common sense knowledge*, and is traditionally acquired from observations or interactions with objects and the environment.

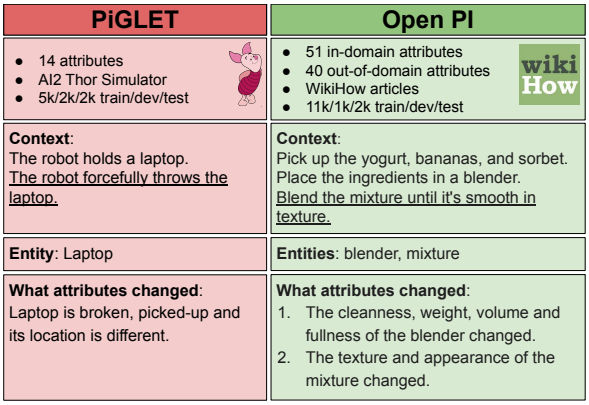

Figure 1: We use the PiGLET and OpenPI datasets to probe if LLMs contain the necessary grounded and world knowledge to reason about event implications.

Core to this line of work is the assumption that events can be learned via language, not depending on other forms of perception. To explore the utility of other modalities and interaction, Zellers et al. (2021) train a language model to predict physical changes in a virtual environment. While intuitively necessary (Bisk et al., 2020a), in this work we show that the purported limitations of language-only models are not always well founded. Key to their success (or failure) are (1) How we use the language models, and (2) The difficulty of the task domain and dataset.

We find that the difficulty of the task is often a stand-in for whether reasoning is required. Others have also noted that despite the tremendous gains

[*] Equal contribution.
[†] Work completed before joining AWS AI Labs.
[1] https://github.com/spilioeve/eventsrealm

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1982 - 1997
December 7-11, 2022

in NLU made possible by Large Language models (LLM), they still stumble when reasoning is required (Brown et al., 2020). If reasoning can be codified as patterns, we are presented with two new challenges: (1) Can we test pattern acquisition via benchmarking generalization, and (2) How can new patterns or context be provided to the model? The nascent field of "prompting" (Liu et al., 2021; Wei et al., 2021; Ouyang et al., 2022) hints at a possible approach for humans to encode novel reasoning patterns for models, however the best structure and the amount of information to convey via a prompt for a given task still remain open questions.

This work makes three contributions to the literature of event implications. First, we show that language by itself provides enough information to predict event implications in current datasets, without the need of a physical interaction model. Second, we establish the difficulty of the problem and limitations of current models by showing extreme differences in performance across different datasets: PiGLET (Zellers et al., 2021), based on a virtual environment, and OpenPI (Tandon et al., 2020), based on procedural text from WikiHow. Third, we explore how different prompting techniques affect model performance in terms of their information content and model nature. Finally, we discuss the generalization properties of our models to unseen attributes (out-of-domain) and how this shows their ability to extract implicit reasoning mechanisms.

2 Related Work

Related work in commonsense follows two directions: (1) predict event implications or track entity changes, and (2) use commonsense knowledge about events and their implications as necessary intermediate steps in reasoning.

Research that directly studies event implications mostly explores causality between social events and emotional states, based on social norm expectations (Rashkin et al., 2018; Sap et al., 2019b; Forbes et al., 2020; Emelin et al., 2020; Hwang et al., 2020). Jiang et al. (2021) study specific linguistic phenomena such as contradiction and negation, while Sap et al. (2019a) study the role of social biases and predicting implications of social events. Although this line of research highlights the difficulty of predicting cause-effect relations, social scenarios are typically ambiguous and require knowledge of event chains. For example, in order to answer whether *X gives a gift to Y* implies that *X hugs Y*, we must be aware of the relation between X and Y, their personalities, and the social context. On the other hand, event implications as physical changes of state of entities are, mostly, objective and depend on simple relations that a model could know a priori (e.g., the material of a mug), allowing us to isolate and study the reasoning abilities of a model.

Closer to our task is the prediction of physical implications of events. This problem often takes the form of entity changes in procedural text, such as in cooking recipes (Bosselut et al., 2017) or WikiHow articles (Tandon et al., 2020). However, most datasets primarily focus on changes in location compared to other attributes, such as ProPara (Mishra et al., 2018) and bAbI (Weston et al., 2015). Modeling approaches in both areas of commonsense explore the generation of explanations in a multi-task setting (Dalvi et al., 2019), the use of external knowledge graph (Tandon et al., 2018), and automatic knowledge base construction to keep a representation of the state of the world and generate novel knowledge (Bosselut et al., 2019; Henaff et al., 2016; Hwang et al., 2020).

The second type of commonsense reasoning includes question answering tasks that assume knowledge of commonsense relations and their implications on the context. This line of work includes short questions, such as OpenBookQA (Mihaylov et al., 2018), CommonSenseQA (Talmor et al., 2019), SWAG (Zellers et al., 2018) and COPA (Roemmele et al., 2011), or questions based on a provided document (Huang et al., 2019) or knowledge base (Clark et al., 2018).

3 Task and Datasets

The problem of predicting event implications can be formulated in several ways, with varying levels of difficulty. For example, Tandon et al. (2020) generate triplets of *entity, attribute, post-state* given some context, while Zellers et al. (2021) are given an entity, attribute, pre-state, and context, to only predict the *post-state* of the entity.

Our task follows a similar formulation to PiGLET, where the model is given a context (i.e., a small paragraph followed by an action-sentence), an entity of interest and a list of attributes. Then, the model needs to determine whether a change-of-state occurred for the entity with respect to the given list of attributes (see Figure 1). However,

unlike Zellers et al. (2021), we do not use the pre-state encoding of the entity, instead we assume that the relevant information is better conveyed through the natural language description of the context.

3.1 PiGLET

PiGLET (Zellers et al., 2021) consists of encodings of the pre- and post-state of entities as a result of an action. Each instance is accompanied by the **context**: a natural language description of the pre-state of the entities, followed by a description of the action. PiGLET is a small dataset (5k training examples), which studies entity change-of-state with respect to 14 attributes, caused by 8 distinct events.

PiGLET is a semi-artificial dataset, where the *entity, pre-state, post-state, action* tuple was generated by exploring the virtual environment AI2 Thor (Kolve et al., 2017). A natural language context was constructed by human annotators, who were provided with the tuples generated by the virtual environment. This results in simpler concise statements compared to the ambiguous language that humans naturally use to communicate.

3.2 OpenPI

Open PI (Tandon et al., 2020) also studies the change-of-state of entities with respect to physical attributes. However, unlike PiGLET, Open PI is based on articles from WikiHow, containing realistic descriptions of physical changes. The context in this dataset is the entire WikiHow article preceding the action sentence from the article.

Open PI is a substantially larger dataset, containing an initial set of 51 pre-defined attributes from WordNet (Fellbaum, 2010), then augmented by human annotators. Although the total number reaches $\sim$800 unique attributes, the initial 51 attributes cover more than 80% of instances. Furthermore, the vast majority of the newly introduced attributes appear only once and many of them contain typos or abbreviations. All our models are trained in the initial set of 51 attributes.

4 Methodology

Next, we introduce our prompting techniques, which vary with respect to per-instance information content. Each technique is tested with different LLMs and fine-tuning methods. The goal of each prompting mechanism is to show how model performance and generalization vary based on the information conveyed in our queries. Our study focuses on four prompting methods depicted in Figure 2: zero-prompt, single-attribute, multi-attribute, and a variant of the latter, the k-attribute prompt.

Our approach builds on literature demonstrating benefits in using prompting to distinguish different tasks, when a model is trained in a multi-task setting (Raffel et al., 2020; Wei et al., 2021). In our study, however, we explore how to use prompts as a medium to convey the task-specific information that a model must know in order to solve the task, similar to how one would ask a human. To the best of our knowledge, we are the first ones to demonstrate advantages and disadvantages of different ways to codify intermediate steps required for reasoning via prompting and use them to study LLMs' understanding of event implications.

4.1 Large Language Models

We explore three transformer-based language models: an autoregressive, an autoencoder, and a seq-to-seq model. We include models with different architectures to investigate the effect of our prompting strategies across model families. Our goal is to use each model in combination with prompts that enhance their individual strengths, based on their pretraining schemes.

RoBERTa (Liu et al., 2019): is an autoencoder model widely used in classification tasks.

T5 (Raffel et al., 2019): is a seq-to-seq model that has shown excellent performance in multi-tasking by using the task description as a prompt. T5 is used for both text classification and generation.

GPT-3 (Brown et al., 2020): is an autoregressive model and is primarily used in zero and few-shot settings due to its substantially larger size. GPT-3 is used in language generation and classification, and has shown excellent performance in few-shot settings when queried with appropriate prompts.

These backbone models are used with one of the three prompting techniques, as described in the following paragraphs and shown in Figure 2.

4.2 Multi-label Classifier: Zero-prompt

Our baseline model is a multi-label classifier with no explicit information about the nature of the task or the attributes themselves. The model takes the context and the prompt *Now what happens next to the [entity]?* as inputs, and predicts a binary vector,

Context: *The robot throws the mug to the ground. What happens next to the mug?*	
Zero-prompt	**Query:** “” **Target:** n-dim binary vector, n = #attributes
Single-attr. prompt	**Query each attribute in candidate list** **Query1:** Is the location of the mug different? **Target:** The location of the mug is different. **Query2**: Is the temperature of the mug different? **Target**: The temperature of the mug is unchanged.
Multi-attr. prompt: *all*-attribute	**Query:** Consider the attributes: location, temperature, shape .... **Target:** The location, composition and shape of the mug changed.
Multi-attr. prompt: *k-attribute*	**Split attributes to subsets** **Query1:** Consider the attributes: location, shape. **Target:** The location and shape of the mug changed. **Query2:** Consider the attributes: temperature, composition. **Target:** The composition of the mug changed.

Figure 2: Prompting techniques used in our models. Multi-attribute prompt improves performance by learning dependencies among attributes.

where entries correspond to changes in specific attributes. We test this mechanism with RoBERTa, as it performs well in classification tasks.

With this model we test the traditional “finetuning assumption” that, given enough data, the model can learn the correspondence between attributes and dimensions in the output vector and correctly predict their changes. This model serves as a baseline of how a LLM performs when fine-tuned to a specific task. Crucially, it does not have the ability to generalize to new attributes as the output vector is of fixed size.

4.3 LM as Classifier: Single-attribute Prompt

Our second prompting technique provides information about individual attributes. Via this technique we evaluate whether a model benefits from the verbalization of each attribute, as a means to retain useful information from the context. Unlike the zero-prompt model, this model can be used out-of-domain, with unseen attributes.

In this setup, we query the model about each individual attribute separately, for every *context-entity* pair, as shown in Figure 2. This mechanism was tested with all three models: RoBERTa (fine-tuned and zero-shot), T5 (fine-tuned) and GPT-3 (few-shot).

By querying each attribute individually, the model is able to focus only on information related to that specific attribute. This can both benefit and hurt performance, as we show in section 5. On one hand, the model pays more attention to the sentence semantics related to the queried attribute. By using the attribute as a bottleneck, the model learns which aspect of meaning is important in that instance. This is particularly beneficial in limited-data scenarios where generalization is necessary. On the other hand, by querying only a single attribute per instance, the model does not learn correlations across attributes. This weakness becomes more apparent in scenarios with many correlated attributes.

4.4 LM as Generator: Multi-attribute Prompt

Our final prompting technique focuses on retrieving information about a set of attributes, by querying multiple attributes together. This technique combines strengths of the zero-prompt and the single-attribute prompt models, as it is able to both verbalize the attributes and capture correlations across them. Unlike other mechanisms, this method allows us to control the information content per instance, by varying the set of queried attributes. As we show in sections 5 and 6.2, varying the attribute queries across training instances is crucial to achieve generalization.

For this technique, the prompt lists the attributes that the model should consider. This list is dataset specific and can vary between training and testing (i.e., out-of-domain) or even across training instances. The model is trained to generate the attributes that changed, as shown in Figure 2. This technique works with text generation models and was tested on both T5 (fine-tuned) and GPT-3 (few-shot).

The first version of this model, the *all-attribute prompt*, queries all attributes that could change in the same instance. However, the risk with this approach is that, because the prompt is fixed, the model learns to pay little attention to the specific attributes that appear in it. We therefore propose a variant of this method, the *k-attribute prompt*, aiming to achieve high performance in both in-domain and out-of-domain scenarios. The objective is to learn about attribute dependencies but also force the model to pay attention to the specific attributes being prompted. To achieve this, we prompt the model with k random attributes and train it to predict changes *only* among these k attributes. More specifically, for each training example, we partition the 51 attributes into q random groups where q is a random integer between 1 and 5. k refers to the number of attributes in each partition. This method ensures that the model is queried with k random attributes and that all 51 attributes are always queried for each example.

Model	All attributes			Per-attribute F1				
	Pr	Re	F1	Dist	Size	Mass	Temp	isBroken
Physical Interaction, (PiGLET)	97.4	91.6	**94.4**	**93.6**	79.2	98.3	**99.6**	92.8
n-gram LogReg (baseline)	87.8	88.0	87.9	78.8	74.7	97.8	94.0	79.4
RoBERTa-base, zero-prompt	95.2	92.6	93.9	90.6	82.7	**100.0**	95.3	**94.7**
T5-base, all-attribute prompt	93.0	95.4	94.1	91.7	**83.5**	**100.0**	95.8	90.3

Table 1: Micro-Precision, Recall and F1 scores across all 14 attributes in PiGLET. Per-attribute F1 scores for challenging attributes, as in (Zellers et al., 2021). Language-only models perform competitively with PiGLET.

5 Experiments & Results

Our task is a multi-label classification where, given some context and an entity of interest, we need to identify which attributes change. Due to the significant label imbalance, in our experiments we report micro- Precision, Recall, and F1 for the positive instances, across labels. In addition to these metrics, we measure per-attribute Precision, Recall and F1 for both datasets (details in subsection A.4).

5.1 PiGLET

Baselines: The strongest baseline is the PiGLET model, which is a combination of physical interaction and language model, based on GPT-2 (Radford et al., 2019). It was proposed in the paper introducing the dataset and is currently state-of-the-art. Unlike the other models, it learns by interacting with a simulator and has access to the pre-state of each entity. We also use a simple n-gram Logistic Regression baseline to both establish the overall difficulty of the dataset and measure benefits due to the pre-training of LLMs.

Results: As shown in Table 1, all models perform relatively well on the PiGLET dataset. The extremely small margin in performance between Physical Interaction and the proposed models (RoBERTa zero-prompt and T5 all-attribute) indicates that language models can learn about physical attributes even without the need of physically interacting with the environment. However, we should highlight that this conclusion holds for datasets similar to PiGLET and the importance of physical interactions remains an open question that must be tested in more realistic and challenging datasets.

Despite the high performance of our proposed models, previously reported baselines on PiGLET show significantly lower performance than the Physical Interaction model. Notably, their baseline using T5-base achieves only 53.9% in hard accuracy, compared to 81.1% of the Physical Interaction model (Zellers et al., 2021). Unfortunately we cannot directly compare these results to our proposed models due to their choice of metric (hard accuracy) and different problem formulation, where the input and output is the encoding of the pre- and post-state of the entity. Despite the use of different metrics, we observe a minimal performance difference between language-only models and PiGLET. This highlights the importance of using proper prompting techniques and task formulation to take full advantage of LLMs and draw valid conclusions.

Our final observation is that there is a larger gap between the n-gram LogReg model and the rest of the models. This shows that, although language is very useful to predict physical event implications, pre-trained language models still have an advantage due to the information they have previously seen. This raises the question of how can we better exploit the relations that pre-trained language models already know, which we explore via the next set of experiments.

5.2 OpenPI

Since our results in PiGLET show that it is not a challenging dataset, we use Open PI to compare the proposed prompting techniques. With the exception of the GPT-3 models, all models have relatively similar sizes, ranging from 123M (RoBERTa-base) to 354M (RoBERTa-large) parameters.

Few-shot: For each instance in the test set, we pick 10 examples from the training set to be included in the prompt - there are marginal improvements beyond four (Min et al., 2022). Performance in complex tasks like QA is sensitive to prompt selection (Liu et al., 2022). Following previous work, we pick the relevant examples based on semantic similarity (Reimers and Gurevych, 2019). In the single-attribute prompt setting, we include examples querying the same attribute, and balance both positives and negatives.

In-domain vs out-domain: All our models are

Training	Model	In-domain Pr	In-domain Re	In-domain F1	Out-domain Pr	Out-domain Re	Out-domain F1
Zero-shot	RoBERTa-large, single-attribute prompt	3.1	63.3	5.9	2.4	68.8	4.6
Few-shot	GPT-3-Babbage, single-attribute prompt	3.7	82.4	7.1	-	-	-
	GPT-3-DaVinci, all-attribute prompt	37.6	24.5	29.7	28.3	12.9	17.7
Fine-tuned	GPT-2 (baseline in Open PI)	49.8	11.8	19.1	-	-	-
	RoBERTa-large, zero prompt	65.1	40.1	49.6	-	-	-
	RoBERTa-base, single-attribute prompt	40.3	55.1	46.6	21.3	26.2	**23.5**
	T5-base, single-attribute prompt	34.6	53.3	42.0	15.9	21.5	18.2
	T5-base, all-attribute prompt	47.5	56.0	**51.4**	25.0	1.2	2.2
	T5-base, k-attribute prompt	52.8	50.0	**51.4**	16.8	22.7	19.3

Table 2: Micro-Precision, Recall and F1 scores for Open PI. In-domain attributes refers to the 51 originally curated attributes, while out-domain to the 41 attributes introduced by human annotators.

trained on the initial 51 attributes (subsection 3.2). For in-domain experiments, the models are tested on the same set of attributes, while for out-of-domain on the new attributes introduced by human annotators. After removal of rare attributes and merging of synonyms, the out-of-domain set consists of 41 unique attributes.

Results: As shown in Table 2, the best performing models in-domain are the multi-attribute prompt models. The performance difference between the multi-attribute models and the zero-prompt baseline shows that the verbalization of attributes has a positive impact on performance, which is further supported by our findings in subsection 6.1. Furthermore, our models beat the GPT-2 model, proposed by Tandon et al. (2020) along with the Open PI dataset. This model generates sentences describing entity state changes but, unlike our models, does not verbalize the attributes. Finally, we observe a drop in performance for both T5 and RoBERTa single-attribute prompt, which confirms that attribute dependencies are important in our task.

Despite its good performance in previously seen attributes, the zero-prompt model cannot classify out-of-domain attributes because its output is a fixed-dimension binary vector. The best out-of-domain performance is achieved by RoBERTa single-attribute, followed by T5 k-attribute prompt.

We observe that, despite the very low out-of-domain performance of the T5 all-attribute prompt model, the other two variants of the same prompting technique (GPT-3 all-attribute and T5 k-attribute) perform competitively. This confirms our hypothesis that fine-tuning with a fixed query hurts the generalization properties of the model, something that can be avoided with few-shot learning or by shifting focus to different attributes during training (i.e., single-attribute or k-attribute).

6 Discussion

We further study the models' behavior with respect to the type of attributes they see and their generalization properties. This analysis serves to uncover advantages and disadvantages of each technique and suggest promising methods for future work to enhance both model performance and robustness.

For all our experiments we use Open PI. Due to its greater diversity of attributes and larger size, it is a better candidate than PiGLET to analyze the limitations of the models.

6.1 Reasoning with Rare Attributes

Since some attributes are significantly more frequent than others, fine-tuned models have been exposed to more data about them, which influences performance. For example, performance across all fine-tuned models for the most frequent attribute *location* is substantially higher compared to other attributes (F1 = 0.65-0.75). Although most models are expected to perform well on such high frequency attributes, our analysis provides useful insights on the models' ability to learn reasoning patterns in limited-data scenarios.

We study per-attribute model performance based on each attribute's frequency in training data for the three prompting techniques: RoBERTa zero-prompt, RoBERTa single-attribute, and T5 all-attribute. After clustering each attribute with respect to its frequency and its F1 score, we ob-

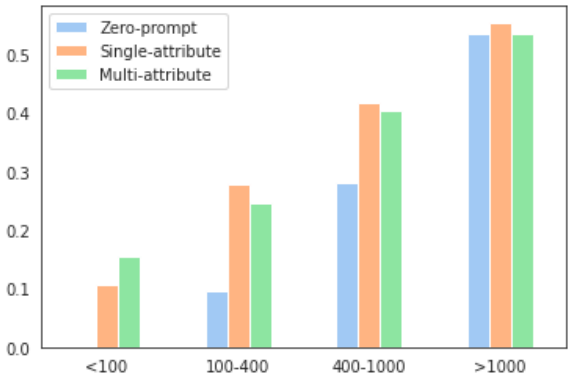

Figure 3: Performance per attribute frequency in training data. Each bar shows the weighted-F1 score across all attributes in the same frequency category.

	RoBERTa, zero-prompt	RoBERTa, single-attribute	T5, all-attribute
Spearman correlation	$\rho = 0.82$	$\rho = 0.80$	$\rho = 0.51$

Table 3: Spearman correlation between attribute frequency and F1 score. High correlation means the model learns primarily high-frequency attributes. All results have p-value < 0.001.

serve four distinct clusters: low (<100 instances), medium-low (100-400), medium-high (400-1000) and high (>1000) frequency. In Figure 3 we plot the weighted-F1 score per cluster for the three models. Our first observation is that performance across all models increases for attributes with higher frequency. This conclusion is also supported by the per-attribute Spearman correlation between performance and frequency, shown in Table 3. This confirms our hypothesis from PiGLET that LLMs can learn physical interactions and achieve higher performance when there is sufficient labeled data to fine-tune on.

Our second observation is that, although performance in high-frequency attributes is similar across all models, it significantly drops for RoBERTa zero-prompt when frequency decreases. This shows that the model struggles to learn with fewer examples. This difference is most striking in the low-frequency cluster, where the model learns nothing (F1 = 0.0). On the other hand, both RoBERTa single-attribute and T5 all-attribute have relatively high performance in low-frequency attributes, where some attributes are easier to learn than others. This supports one of our main hypothesis in this paper that, by *verbalizing and querying specific attributes, models pay attention to each attribute and learn reasoning patterns, a crucial step in limited-data scenarios.*

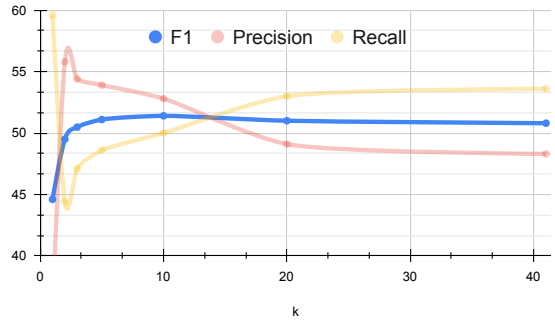

Figure 4: F1, Precision, and Recall scores as a function of the number of attributes used in the prompt during evaluation for the k-attribute model

6.2 Prompt Diversification via the k-attribute Prompt Model

Through manual inspection we find that the all-attribute models have an inherent bias towards generating attributes that appeared in the training data, even when prompted with new ones. Their performance is in fact poor in the out-of-domain setting (2.2 F1, Table 2). Now the question is whether this is a limitation of the reasoning abilities of the multi-attribute models or a bias introduced by its training scheme.

We propose the k-attribute model to alleviate training biases by randomizing the queried attributes. Notably, this model still maintains the core assumptions behind the multi-attribute prompt model of querying multiple attributes at once. We observe that this simple technique results in the same in-domain F1 score as the all-attribute prompt model, while significantly improving its out-of-domain performance. This shows that the observed limitations with the all-attribute prompt model are due to training biases that prevent the model from generalizing to unseen attributes.

Once trained, the k-attribute prompt model can be queried with varying number of attributes. In Figure 4, we plot the performance of the model as a function of the number of attributes used in the query during evaluation. We observe a drop in performance when the model is queried with a single attribute (similar to the single-attribute prompt models). The performance is highest around 10 attributes and drops slightly beyond that. We also observe that by varying k, we can modulate precision and recall, suggesting that there are both

lower and upper bounds on the optimal number of attributes that LLMs can consider at once.

We also experimented by grouping attributes in a prompt based on their semantic similarity, but this did not yield any significant changes in performance. We leave it to future work to investigate further how to optimally choose the groups to use in a prompt during training and inference.

6.3 Semantic Similarity and Generalization

A major obstacle for NLP models is to apply the reasoning patterns they have learned to unseen attributes. Although the overall performance is lower in out-of-domain (best F1 = 23.5) compared to in-domain experiments (best F1 = 51.4), we observe that it varies significantly across different attributes. In this part of our analysis, we investigate the models' generalization abilities to out-of-domain attributes, based on their relation to in-domain attributes.

Essentially we identify two types of out-of-domain attributes: (1) these that are semantically similar to some in-domain attribute(s), and (2) these that have no similarity to any in-domain attribute. These two groups of attributes also evaluate the degree of the model's generalization abilities, as it is easier to generalize to different verbalizations of a previously seen attribute than to a completely new concept. For this part of the analysis we use the RoBERTa single-attribute prompt model, as it has the best out-of-domain performance.

To identify related attributes, we firstly use cosine similarity distance on top of an encoder trained for semantic similarity (Reimers and Gurevych, 2019). After manual curation, we identify 21 out-of-domain attributes that are closely related to in-domain attributes (Group Matched), as we see in Table 7. The 20 remaining out-of-domain attributes are more dissimilar and do not have matching in-domain attributes (Group Dissimilar).

For each of the two groups (Group Matched and Group Dissimilar), we estimate the weighted-F1 score. We observe that Group Matched reaches **F1 = 29.4**, while Group Dissimilar **F1 = 13.6**. For Group Matched, we also verify that the model's performance on closely related attributes is similar by measuring their Pearson correlation, which is $r = 0.67$ (p-value < 0.05). Both results indicate that *the model understands the semantics of the attributes despite different verbalizations, however, it struggles with more complex reasoning mechanisms, such as applying the acquired patterns to entirely new attributes*.

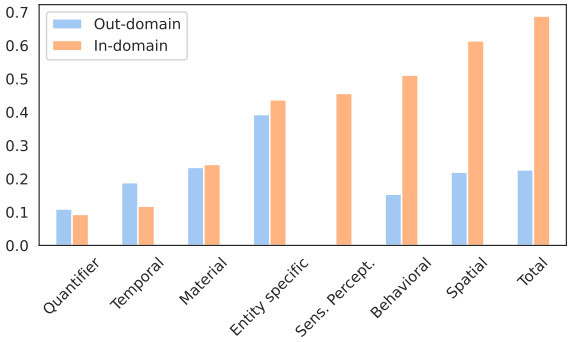

Figure 5: F1 scores per attribute semantic type.

6.4 Challenging Semantic Types

In this part of our analysis, we explore why some classes of physical attributes appear to be inherently more difficult for LLMs. More specifically, we manually design an ontology of attributes into seven major semantic types and then group each in-domain and out-of-domain attribute according to the information it encodes, as seen in subsection A.6. Via this analysis we aim to identify evaluate each semantic type with respect to: (1) in-domain performance, and (2) generalization to unseen attributes. For this analysis we use RoBERTa single-attribute prompt, as it has the best out-of-domain performance.

Figure 5 shows that the model particularly struggles to predict attributes of the *Quantifiers* and *Temporal* semantic types (in-domain). These attributes are known to be challenging for current LLMs (Ravichander et al., 2019).

We further observe that the *Entity-specific* and *Material* semantic types are equally challenging for both in-domain and out-of-domain attributes. These semantic types describe inherent properties of an entity, such as *fullness*, that can only change due to very specific events, such as *put X into Y*. On the other hand, the *Spatial* and *Behavioral* types show a large discrepancy between in-domain and out-of-domain performance. This is surprising given that these semantic types contain high-frequency attributes, like *location*. This highlights *the limitations of current models to predict physical changes outside of controlled environments*.

6.5 Error Analysis

To identify the cause of low out-of-domain performance and study the models' generalization abil-

ities, we perform a manual error analysis of out-of-domain outputs from the best performing models: T5 k-attribute and RoBERTa single-attribute prompt.

We identify four major types of errors indicating a varying degree of understanding of context and entities involved. The results of this analysis are shown in Table 4.

False negatives: correct predictions that are missing from the annotations. This error type does not reflect a failure of the models, but rather of the dataset which was crowd sourced. Since out-of-domain attributes were introduced by workers on Amazon Mechanical Turk, each annotator may introduce attributes that were not considered by others while annotating different instances. This is particularly prominent among similar concepts, such as *width* and *size*, which oftentimes change together. As we see in Table 4, *false negatives* are responsible for 41.5% of errors made by T5 k-attribute prompt and 25.4% of those made by RoBERTa single-attribute. This highlights that the gap between out-of-domain and in-domain performance is narrower than what our automated evaluation showed.

False negative errors can be divided into two subcategories. The first category accounts for predicted attributes that are synonyms of the annotated attributes and could replace them in the particular instance. The second category comprises predicted attributes that significantly differ but complement the annotated attributes, such as *flexibility* and *size*. We found that the first category of synonyms is responsible for 53% (T5 k-attribute prompt) and 44% (RoBERTa single-attribute) of the instances with *false negative* errors.

Wrong context: predictions that could be correct for the given entity, but incorrect given the context. This error represents the models' challenges with respect to event implications and reasoning.

Wrong entity: wrong attribute change predictions for the given entity in any context. This is the most severe error since it shows that the model is not able to link the attributes to the entity. While this error is very rare for the T5 k-attribute model (only 2.7%), it is frequent for the RoBERTa single-attribute model (20.7%).

No prediction: instances with null predictions. This is the most frequent error type for both models, accounting for almost half of the errors. This error occurs when the model decides that there is no attribute change from the given list of attributes, which results in a significant drop in recall. This highlights that both models struggle to identify which out-of-domain attributes are relevant to a particular context and entity.

Error Type	**T5 k-attribute**	**RoBERTa single-attribute**
False negatives	41.5%	25.4%
Wrong context	7.6 %	6.5%
Wrong entity	2.7%	20.7%
No prediction	48.2%	47.4 %

Table 4: Error categories and prevalence of each category as a percentage of the number of instances. Based on out-of-domain attributes. *Wrong context* implies the prediction could be correct for the given entity but is incorrect in the given context. *Wrong entity* means the attribute change does not apply to the given entity in any context.

7 Conclusion

Predicting physical changes due to events is a challenging problem for current models, especially in out-of-domain or limited-data scenarios. We show that, by using proper task formulation, LLMs can learn physical event implications even without physical interactions. Future work should explore the question of whether physical interactions are necessary in more complex and realistic settings, by (1) providing more challenging datasets that test the model limitations, and (2) ensure a fair comparison of the language-only baselines.

Furthermore, we show that the performance of a LLM may significantly vary based on how we use it, and, overall, LLMs can benefit from: (1) verbalizing the attributes, (2) varying the prompt information content across instances, and (3) querying multiple attributes in the same instance. By following these guidelines, we show significant improvements in unseen attributes and attributes of low-frequency. Last, our error analysis and discussion sections provide useful insights for future work, with respect to prompt content and shortcomings of the current datasets that study physical event implications.

8 Acknowledgments

We thank the anonymous reviewers for their helpful feedback. We also thank Alan Ritter and Lori Levin for their comments and feedback.

9 Limitations

Computing resources The different prompting methods have trade-offs in terms of computational costs. In particular, the all-attribute and zero-attribute query all changes at once. With the k-attribute prompt, we query attributes in smaller groups requiring on average #attributes/k times more computations than for the all-attribute model (in our case five times). The single-attribute model encodes each attribute separately requiring #attributes times more computations. We were unable to test GPT-3 for single-attribute because of the cost of the larger number of queries it would have required. The experiments that did not involve GPT-3 were run on two NVIDIA K-80 GPUs with 12Gb memory.

Dataset limitations Given the complex nature the event implication task, both datasets have several limitations. PiGLET, which is based on a virtual environment, has relatively simple language that is not representative of naturally occurring text. Furthermore, because it is a relatively small dataset with respect to number of attributes and entities, the training set covers a large subset of the possible configurations in that virtual environment. This explains the very high performance of all models.

Although Open PI does not suffer from such limitations, we discovered several inconsistencies in the annotations. These inconsistencies mainly involve: (1) wrong attributes, (2) inconsistent labeling, and (3) duplication of attributes. Although we manually edited several of these problems by merging and filtering attributes, we could not address the inconsistencies in labeling. This resulted could have influenced model performance.

Automatic Prompt Generation In this work, we did not explore whether prompts can be automatically generated. There have been several recent studies aiming at generating either discrete or soft prompts (Shin et al., 2020; Lester et al., 2021). In our case, the changes in information content involved a deeper understanding of the task and required human involvement. As the field of prompt generation matures, future work could investigate automating the process of finding prompts with variable information content.

Multi-task learning We do not directly explore benefits from multi-task learning even though Raffel et al. (2020); Wei et al. (2021) show that this can significantly improve zero-shot and few-shot performance. However, the GPT-3 model that we used in our experiments is the Instruct GPT-3 model which is the result of additional prompt-based fine-tuning.

References

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. 2019. Phyre: A new benchmark for physical reasoning.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020a. Experience Grounds Language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020b. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

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.

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

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wentau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. 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 4496–4505.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*.

Denis Emelin, Ronan Le Bras, Jena D Hwang, Maxwell Forbes, and Yejin Choi. 2020. Moral stories: Situated reasoning about norms, intents, actions, and their consequences. *arXiv preprint arXiv:2012.15738*.

Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Maxwell Forbes, Jena D Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. *arXiv preprint arXiv:2011.00620*.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2020. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. *arXiv preprint arXiv:2010.05953*.

Liwei Jiang, Antoine Bosselut, Chandra Bhagavatula, and Yejin Choi. 2021. " i'm not mad": Commonsense implications of negation and contradiction. *arXiv preprint arXiv:2104.06511*.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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*.

John McCarthy and Patrick J Hayes. 1981. Some philosophical problems from the standpoint of artificial intelligence. In *Readings in artificial intelligence*, pages 431–450. Elsevier.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint*.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939*.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2019a. Social bias frames: Reasoning about social and power implications of language. *arXiv preprint arXiv:1911.03891*.

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

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Niket Tandon, Bhavana Dalvi Mishra, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. *arXiv preprint arXiv:1808.10012*.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Rowan Zellers, Ari Holtzman, Matthew E Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. 2021. Piglet: Language grounding through neuro-symbolic interaction in a 3d world. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2040–2050.

A Appendix

Our experiments are built on top of the Huggingface library (Wolf et al., 2019).

A.1 Metrics

Our task is a multi-label classification where, given some context and an entity of interest, we need to identify which attributes change. For most pairs *context, entity*, event implications affect only 1-2 attributes. This results in a few positive instances (i.e., attributes that change) and a large number of negative instances (i.e., attributes that do not change). Furthermore, we observe that the number of positive instances significantly varies across attributes: for example, in the training set of Open PI, *location* has 4505 positive instances, while *distance* only 53. Due to the significant label imbalance, in our experiments we report micro- Precision, Recall, and F1 for the positive instances, across labels. In addition to these metrics, we measure per-attribute Precision, Recall and F1 for both datasets.

A.2 Hyperparameters

We performed hyperparameter search in the following way. Based on the model size, we picked the largest batch size that could fit on our GPUs. Then we performed hyperparameter search on the dev set (6 values in range $[10^{-3}, 10^{-6}]$), label smoothing (0, 0.1, 0.2) via grid search. We report in Table 5 the hyperparameters we use in each case. We used the default values in the transformer library for the rest. For T5 we also varied the task prefix and its position based on the relevant pre-training tasks, without observing significant differences. We use Adam with betas (0.9,0.999) and ϵ =1e-08 for T5 experiments. The runtime for each hyperparameter combination in Open PI is: about 2 hours for multi-attribute, about one hour for zero-prompt, about two days for single-attribute (T5 and RoBERTa have similar runtime).

Data	Model	Epochs	Batch size	Learning Rate	Label Smoothing
PiGLET	RoBERTa, zero-prompt	30	20	4e-05	0.0
	T5 all-attr	50	32	3e-05	0.1
Open PI	RoBERTa, zero-prompt	20	32	1e-05	0.0
	RoBERTa, single-attr	6	16	1e-05	0.1
	T5 single-attr	8	16	5e-05	0.1
	T5 all-attr	8	16	5e-05	0.1
	T5 k-attr	10	16	5e-05	0.1

Table 5: Hyperparameters

To verify that model size differences do not impact our results, we also did experiments with RoBERTa-base zero-prompt, which shows very similar performance to RoBERTa-large zero-prompt.

A.3 In-domain Attributes and their Frequency

Attribute	Train	Dev	Test
location	4505	360	803
cleanness	1255	117	167
wetness	1211	80	215
temperature	1184	91	184
weight	1073	84	124
fullness	694	62	122
volume	676	56	174
composition	662	48	90
shape	538	55	65
texture	515	34	74
knowledge	409	27	119
orientation	330	15	45
color	292	13	33
size	264	26	50
power	245	11	18
organization	242	14	37
motion	242	15	33
ownership	212	6	19
availability	195	30	63
step	171	8	13
speed	151	3	18
pressure	148	4	14
taste	145	8	14
length	122	9	17
electric conductivity	121	9	18
smell	120	7	43
sound	68	6	6
brightness	65	0	7
thickness	64	4	16
strength	64	2	14
hardness	63	5	10
skill	62	3	4
openness	55	2	16
coverage	54	3	7
stability	54	6	14
focus	53	4	5
cost	53	6	9
distance	53	0	11
appearance	44	8	8
complexity	44	1	5
amount	40	3	16

Table 6: Attribute occurrences in training, validation, and test sets.

A.4 In-domain performance, per-attribute

In Figure 6 we show the in-domain F1 score per attribute for RoBERTa zero-prompt and T5 multi-attribute prompt models in Open PI. The attributes are sorted according to their frequency (decreasing).

We observe that RoBERTa zero-prompt completely ignores all attributes with less than 150 instances. Furthermore, the only attributes that RoBERTa zero-prompt performs better are *location, cleanness, temperature, size* and *power*. Although for 4/5 of these attributes the difference in F1 score between the two models is marginal, the fact that 3/5 belong to the most frequent attributes (more than 1000 instances) influences the overall micro-F1.

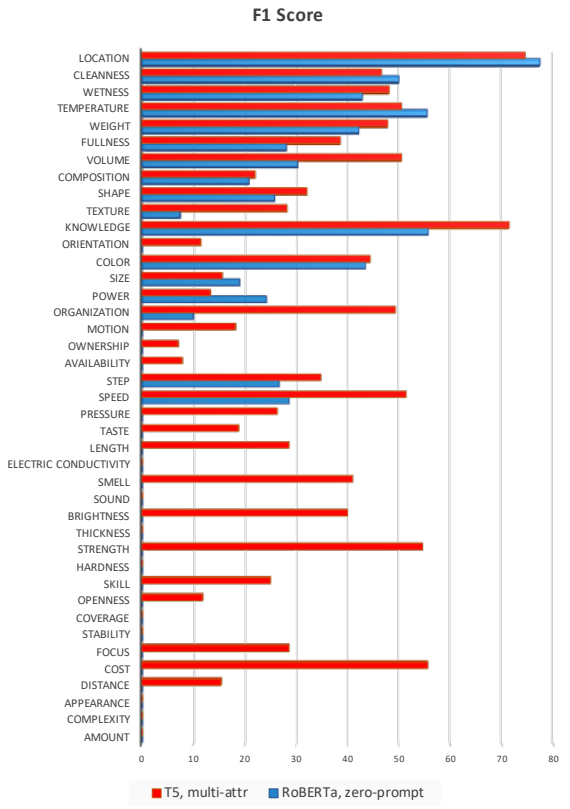

Figure 6: F1 score per attribute for RoBERTa zero-prompt and T5 multi-attribute prompt models in Open PI.

A.5 Semantically Similar Attributes

In Table 7 we show for every out-of-domain attribute, the most semantically similar in-domain attribute. This list contains only out-of-domain attributes that had a synonym from the in-domain group (Group Matched). This group was formed after manual inspection of the automatically generated synonym pairs.

Out-of-domain attribute	**In-domain synonym/antonym**
activity	motion
angle	orientation
area	shape
balance	weight
capacity	amount
consistency	stability
contents	composition
direction	orientation
flexibility	stability
granularity	composition
height	length
hydration	wetness
intensity	brightness
quantity	amount
safety	speed
softness	hardness
tenseness	pressure
tension	pressure
thermal conductivity	electric conductivity
tightness	pressure
width	length

Table 7: The most semantically similar in-domain attribute, each out-of-domain attribute.

Semantic Cluster	In-domain Attributes	Out-of-domain Attributes
Spatial	location, volume, shape, orientation, size, length, distance, organization	angle, direction, area, height, width, pose, posture, spacial relation
Material	texture, electric conductivity, thickness, hardness, strength, pressure	tenseness, tension, tightness, softness, material, flexibility, thermal conductivity, density, granularity
Entity-Specific	cleanness, wetness, fullness, ownership, openness, cost, composition, coverage, focus	contents, wholeness, capacity, hydration, consumption, documentation, emotional state, pain, usage
Behavioral	knowledge, speed, motion, stability, complexity, skill	activity, balance, consistency, safety, familiarity, exposure, viability, resistance
Quantifier	amount	intensity, quantity, magnitude
Temporal	availability	age, life, existence, time
Sensory Perception	visibility	color, taste, temperature, smell, sound, appearance, weight, brightness

Table 8: Semantic clusters of attributes, both in-domain and out-of-domain.

A.6 Semantic Clusters of Attributes

Table 8 shows the semantic clusters of attributes which are the result of agglomerative clustering and manually curation of in-domain and out-of-domain attributes. These clusters help better understand our attributes and performance based on their semantics. The clusters were used in Section 6.3.

A.7 OpenPI Real Examples

Examples from out-of-domain with model predictions from the T5 k-attribute prompt and the RoBERTa single-attribute prompt models. In many instances the predicted attribute is correct, but the annotations fail to reflect this.

In Table 9, we show some real instances that we used in our error analysis. Although for each instance all the out-of-domain attributes were queried, for brevity we only show attributes that were identified as changed by either model or by the annotations. We observe that in many of these examples the models predict attribute changes that are correct, despite not being captured by the annotations. Such cases are Example 2, Example 4 and Example 5, where the T5 k-attribute prompt correctly predicts attributes that were not identified by the annotators. These attributes are not necessarily related to the annotated attribute, such as *width* and *resistance* in Example 2, or *hydration* and *softness* in Example 4. However, some other instances may have predicted attributes that are closely related to the annotated attribute, as we see in Example 1, where *posture* and *angle* oftentimes change together.

Our final observation from Table 9 is that the models are able to correctly predict attributes that require some common sense knowledge, which was not part of the provided context. For example, T5 k-attribute prompt predicts in Example 4 that *soaking beans* implies that *softness* changes, something that is not as an obvious conclusion as the change of *hydration*. Even more, in Example 5 we observe that the model is able to understand the intent of the paragraph, which is to change the *softness of lips*. These examples show that the T5 k-attribute prompt model is able to perform some degree of reasoning, even for predictions that were considered wrong due to missing annotations.

Example 1	
Context: Begin by standing in Mountain Pose. Bend your right leg back and hold on to the inside of your foot behind you with your right hand.	**Entity:** person **Annotated Attributes:** balance
T5 k-attribute prompts: Consider the following attributes: flexibility, angle, hydration, consumption. Which attribute changed for the person?	**T5 k-attribute output:** posture, flexibility, angle, pose
RoBERTa single-attribute prompts: Is the flexibility of the person different? Is the viability of the person different?	**RoBERTa single-attribute output:** No Yes
Example 2	
Context: Cut off a corner of a yeast packet.	**Entity:** packet **Annotated Attributes:** resistance
T5 k-attribute prompts: Consider the following attributes: contents, angle, width, resistance, softness. Which attribute changed for the packet?	**T5 k-attribute output:** contents, width
RoBERTa single-attribute prompts: Is the width of the packet different? Is the resistance of the packet different?	**RoBERTa single-attribute output:** Yes No
Example 3	
Context: Drink a glass of hot milk.	**Entity:** body **Annotated Attributes:** thermal conductivity
T5 k-attribute prompts: Consider the following attributes: contents, hydration, thermal conductivity. Which attribute changed for the body?	**T5 k-attribute output:** thermal conductivity
RoBERTa single-attribute prompts: Is the thermal conductivity of the body different? Is the hydration of the body different?	**RoBERTa single-attribute output:** No Yes
Example 4	
Context: Soak the dried beans and lentils overnight in a large bowl.	**Entity:** beans **Annotated Attributes:** hydration
T5 k-attribute prompts: Consider the following attributes: softness, contents, granularity, hydration. Which attribute changed for the beans?	**T5 k-attribute output:** softness
RoBERTa single-attribute prompts: Is the hydration of the beans different? Is the softness of the beans different?	**RoBERTa single-attribute output:** No No
Example 5	
Context: Take the honey and mix it with the sugar, then add in a little bit of Vaseline or petroleum jelly. When the mixture is all gritty, apply it on to your lips as you would with lip balm. Leave on the mixture for about one minute.	**Entity:** lips **Annotated Attributes:** granularity
T5 k-attribute prompts: Consider the following attributes: softness, pain, granularity. Which attribute changed for the lips?	**T5 k-attribute output:** softness, pain
RoBERTa single-attribute prompts: Is the softness of the lips different? Is the granularity of the lips different?	**RoBERTa single-attribute output:** No No

Table 9: Examples from out-of-domain and model predictions for the T5 k-attribute prompt and the RoBERTa single-attribute prompt models.