PaCo: Preconditions Attributed to Commonsense Knowledge

Anonymous ACL submission

Abstract

Humans can seamlessly reason with circumstantial preconditions of commonsense knowledge. We understand that *a glass is used for drinking water*, unless the glass is broken or the water is toxic. Despite state-of-the-art (SOTA) language models' (LMs) impressive performance on inferring commonsense knowledge, it is unclear whether they understand the circumstantial preconditions. To address this gap, we propose a novel challenge of reasoning with circumstantial preconditions. We collect a dataset, called *PaCo*, consisting of 12.4 thousand preconditions of commonsense statements expressed in natural language. Based on this dataset, we create three canonical evaluation tasks and use them to examine the capability of existing LMs to understand situational preconditions. Our results reveal a 10-30% gap between machine and human performance on our tasks, which shows that reasoning with preconditions is an open challenge. Upon acceptance, we will release the dataset and the code used to test models.

1 Introduction

Improving a system's ability to reason with commonsense knowledge is at the frontier of natural language processing (NLP) research, as a critical component in many knowledge-driven tasks such as question answering (Wang et al., 2019; Talmor et al., 2019), machine reading comprehension (Sakaguchi et al., 2020), narrative cloze (Mostafazadeh et al., 2016), and dialogue systems (Adiwardana et al., 2020; Young et al., 2018). Recently, dozens of systems (Raffel et al., 2019; Khashabi et al., 2020; Liu et al., 2019; Devlin et al., 2019) and learning resources (Sap et al., 2019b; Mostafazadeh et al., 2020; Rudinger et al., 2020; Bhagavatula et al., 2020) have been proposed, focusing on various aspects of commonsense knowledge such as naive physics and naive psychology.

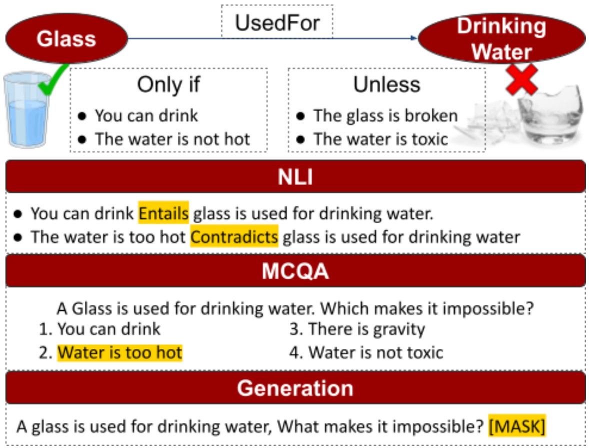

Figure 1: Overview of the *PaCo* data collection and instances of the three tasks derived from it.

In cognitive studies, the *theory of affordance* (Gibson, 2000; Chemero, 2003) suggests that understanding the circumstances in which an action or statement is possible or impossible is a key aspect of human intelligence. For example, a glass may be used for drinking water, under an implicit assumption that the water is at normal temperature, but may not if the glass is shattered. Accordingly, we argue that for an NLP reasoner to understand common sense, it should comprehend the contextual *preconditions* associated with commonsense statements. Such contextual preconditions can naturally be categorized into two classes: the ones that *enable* the statements, and the ones that *disable* them (Fikes and Nilsson, 1971; Hobbs, 2005).

Causal preconditions may be partially inferred from text (Mostafazadeh et al., 2020; Kwon et al., 2020), however: 1) as is the case in many other aspects of common sense, we rarely write them explicitly in our text; 2) when mentioned in the text, it is difficult for models to distinguish whether they represent causation or correlation. Similar to our work, Rudinger et al. (2020) collect the preconditions by crowdsourcing. Here, the preconditions are seen as soft assumptions, namely: *weakeners* and *strengtheners*, which provides a model only with the relative correlation between statements, and is not explicitly testing the model on the underlying preconditions of the statement. Instead, we propose to define the problem based on the crisp conditioning of *disablers* and *enablers*, which forces the LM to learn the decisive preconditions of a statement and facilitates explainability based on them. In comparison to a hard logical connection modeled by the crisp condition, although the notion of *weakener* is also helpful to the commonsense reasoner, it raises additional questions like "by how much?", or "is the statement still valid?". Whereas in the notion of *disablers*, even though annotations are more difficult to collect, it can at least take the system one step forward by sorting out the clutter of the irrelevant statements.

This paper presents a systematic study on the problem of situational preconditions expressed in natural language. As the **first** contribution, we define a new problem of reasoning with *enabling* and *disabling* preconditions associated with commonsense statements (Section 2). Given a statement, the task is to infer the preconditions that make the statement possible (*enabling*) or impossible (*disabling*). Understanding such preconditions of commonsense knowledge would enable reasoning systems relying on a commonsense knowledge base to decide when to use a given commonsense statement. For example, given the statement "Glass is used for drinking water" in ConceptNet (Speer et al., 2017), a system should know that it is only possible if the "water is not too hot", and it is impossible when "the water is toxic".

To foster research on preconditions of commonsense knowledge, we develop *PaCo*, a rich crowdsourced dataset with *enabling* and *disabling* preconditions of commonsense statements (Section 3), as the **second** contribution of this paper. For *PaCo*, we start by extracting available commonsense statements. We then design and execute a crowdsourcing task to gather preconditions of the statements by asking participants: *what makes the statement possible/impossible?* for each of the statements. *PaCo* contains 12.4K labeled preconditions (6.6K *enabling*, 5.8K *disabling*), corresponding to $3 * 1$K edges from three representative relations in ConceptNet (Speer et al., 2017), covering knowledge on utility, causality, and motivation. Example preconditions are illustrated in Fig. 1. These tasks for the first time allow analysis beyond what is done in prior work that cover enabling preconditions only. Particularly, they realize a head-to-head comparison of enabling and disabling statements which was not possible before. Besides, they allow analysis of the impact of the knowledge types (e.g., utility) on the task difficulty for both humans and neural language models.

Our **third** contribution is an extensive NLP benchmarking based on *PaCo*. To this end, we transform *PaCo* into three tasks on **P**reconditions: **N**atural **L**anguage **I**nference (P-NLI), **M**ultiple-**C**hoice **Q**uestion **A**nswering (P-MCQA), and **G**eneration (P-G). The three canonical tasks seek to provide a comprehensive evaluation of the ability of natural language reasoners to understand circumstantial preconditions (Section 4). These three tasks examine the understanding of preconditions of a number of SOTA language models and reasoners, such as DeBERTa (He et al., 2020), and UnifiedQA (Khashabi et al., 2020). Results show that SOTA methods largely fall behind human performance, therefore indicating the need for further research in order to improve the comprehension of contextual preconditions by commonsense reasoners (Section 5).

2 Preconditions in Commonsense Reasoning

Problem Definition. Commonsense statements describe well-known information about concepts, and, as such, they are acceptable by people without need for debate (Sap et al., 2019a; Ilievski et al., 2020b). A commonsense statement can be formalized as $s = (h, r, t)$, where h and t are head and tail concepts, and r is the relation type.

Following the notion of "causal complex" (Hobbs, 2005), we define the precondition P_f as a collection of eventualities (events or states) that results in s to happen. Such preconditions contain eventualities that either *allow* ($p_f^+ \in P_f$) or *prevent* ($p_f^- \in P_f$) the statement to happen. Here, to *prevent* means to *allow* the negation of the statement (Fikes and Nilsson, 1971). While enumerating a priori all such causal eventualities is impossible, people are still able to reason about them in a given situation (Hobbs, 2005). Notably, preconditions are *implicit*, i.e., we usually omit them from conversation as they are considered obvious (Grice, 1975). Shoham (1990) and Hobbs (2005) distinguish between two type of preconditions, based on causal connections (*hard*), or material implication (tends to cause; *soft*). Here we focus on the more restrictive, *hard* preconditions; for soft preconditions, see (Rudinger et al., 2020).

In this work, the problem of reasoning with preconditions is attempted in two ways: discriminative and generative (cf. Table 1). In the discriminative setting, given a statement f and a precondition (p), a model is expected to infer if the fact is still valid ($p \in P_f^+$) or not ($p \in P_f^-$). In the generative setting, given only the statement (f), a model is requested to compose a reasonable disabling (p_f^-) or enabling (p_f^+) precondition.

Motivating Examples. In a preliminary investigation, we assess the ability of SOTA language models: GPT2 (Radford et al., 2019), and UnifiedQA (Khashabi et al., 2020), to reason with preconditions. As shown in Table 1, both models appear to fall short of reasoning with enabling and disabling factors of commonsense statements, regardless of whether the prompt task form is presented as multiple-choice question answering (row 1), or as text completion (rows 2-4). This observation is not surprising, considering that reasoning with preconditions is an under-addressed research challenge. Yet, it motivates the urgency for this problem to be studied in depth, which is the goal of this paper.

3 *PaCo*

This section introduces the procedure of developing the *PaCo* dataset. We start by selecting relevant commonsense facts (Section 3.1), and crowdsourcing preconditions for each statement (Section 3.2). Finally, we present the *PaCo* data statistics (Section 3.3).

3.1 Edge Selection

We extracted relevant commonsense facts from ConceptNet (Speer et al., 2017). We chose ConceptNet due to its breadth of knowledge and popularity in prior research (Feng et al., 2020; Lin et al., 2019; Ma et al., 2019). ConceptNet is a publicly available common sense knowledge resource. It contains 3.4 million English assertions between concepts (e.g., “Glass”, “Drinking_water”, “Person”), and covers a wide range of knowledge types, including spatial, physical, and temporal knowledge, as well as social and cognitive knowledge about everyday situations.

We performed a pilot analysis of different knowledge types in ConceptNet to help us decide which of them were suitable to be annotated with preconditions. Namely, we sampled 20 random edges for each relation and checked how well one could annotate them with preconditions. Our analysis revealed that not all relations lent themselves naturally for annotation with enabling or disabling preconditions. Specifically, we observed that some relations (e.g., *Related To*) are underspecified in their meanings, and others, like *IsA*, are often truisms. Our investigation has revealed that it is difficult to come up with preconditions for these relations. Furthermore, we observed that some relations, like *CreatedBy*, could be easily annotated with enabling conditions, but not with disabling ones. The opposite was observed for *PartOf*.

We opted for the relations *UsedFor*, *Causes*, and *Desires*, because of their suitability for annotation of preconditions, their relatively high number of statements, and their representativeness of three different dimensions of knowledge: utility, temporal, and motivational knowledge (Ilievski et al., 2021). Following the intuition that not all statements can be annotated with preconditions, e.g., *(Looking through telescope, Usedfor, viewing heavens)*, we computed the correlation between a hand-annotated suitability judgment for the precondition statements, and the several quantitative scores: DICE metrics (Chalier et al. 2020; e.g., salience), LM perplexity, and edge weights in ConceptNet. However, none of these scores had a strong correlation with the suitability for annotating preconditions (Appendix B.1 contains the calculated correlations for *UsedFor*). Therefore, we opted for the relations *UsedFor*, *Causes*, and *Desires*, because of their suitability for annotation of preconditions, high number. Also they are representative of three different dimensions of knowledge: utility, temporal, and motivational knowledge (Ilievski et al., 2021). We sampled 1K edges from each and lexicalized them into human readable sentences using relation-specific templates (see Appendix A.4).

3.2 Data Collection

Mechanical Turk We used Amazon Mechanical Turk (Crowston, 2012) to collect data on preconditions for the lexicalized statements as part of Institutional Review Boards (IRB) approved (as exempt) study. For this, we asked the participants to provide short responses to the question: “What makes the statement possible/impossible?” for each of the lexicalized statements from ConceptNet. Due to fi-

Model	Input	Output
UnifiedQA	A net is used for catching fish. What makes this impossible? (A) You are in water (B) You are in downtown LA	You are in water
UnifiedQA	A net is used for catching fish. What makes this impossible?	A net is used for catching fish.
GPT2	A glass is used for drinking water only if, the glass	is covered in a protective coat or can be removed with cold water.
GPT2	A glass is used for drinking water only if, the water	is acidic, not fresh.

Table 1: Test of language model's understanding of preconditions

nancial limitations, we restricted our annotations to 3 enabling and 3 disabling judgments for each statement. While the goal of *PaCo* is not to exhaust all possible preconditions associated with each statement, for some statements we observed duplicate answers, signaling a near-saturation point.

Further details on the data collection design, including annotator qualification, and survey design details are given in Appendix A. With this procedure, we collected a total of 18K enabling and disabling preconditions.

Quality Control We use a mixture of automated and expert annotations for quality control. The automated quality control consisted of three rules that we can programmatically check: 1) not using negative words like "not", 2) not using pronouns, and 3) proper sentence lengths. In order to measure the informativeness and relevance of the remaining annotations, we use expert annotation. Specifically, for a subset of the recorded responses we asked the annotator to classify the response into three categories, each representing a specific level of informativeness in the response: 1) *Truism*: the response is correct, but it is not specific to the situation (e.g., *being broken/functional* or *being available/unavailable*); 2) *Informative*: the response is correct and is adding information that is not mentioned in the prompt, while not being a truism (i.e., is specific); 3) *Irrelevant*: any response that is not placed into the previous two categories. For *PaCo*, we remove the answers from the *Irrelevant* category, while truism answers could be removed subsequently if so desired.

3.3 Dataset Statistics

This data collection procedure resulted in a total of 9k enabling and 9k disabling preconditions for each of the 1k ConceptNet edges selected for *UsedFor*, *Causes*, and *Desires* relations respectively. After filtering out responses in low quality and those marked as *Invalid* by crowd annotators, *PaCo* contains 12.4K annotations (6.6K *enabling*, 5.8K *disabling*). Our expert annotation on 10% of the 6K annotations with *UsedFor* relation showed that in 93% of the crowdsourced responses are informative, whereas only 5% of the responses are irrelevant. The quality of the responses is lower for the two other relations: 70% informative responses for *Causes* and 61% for *Desires*. This shows that the two relations are semantically more challenging to human annotators compared to a utility relation like *UsedFor*. We also observed that on average it took the annotators 3.5 times longer to submit a responses for these two relations, which confirms that *UsedFor* is the most suitable of the three relations for associating preconditions.

ID	Instance
P-NLI	*Hypothesis: A net is used for catching fish* *Premise: We are in a desert* *Label: Contradiction*
P-MCQA	*Question: A net is used for catching fish. When is this impossible?* *Choices: (A) You are in sea, (B) The boat is moving, (C) Net has a large hole in it.*
P-G	*Question: A net is used for catching fish. When is this impossible?* *References: (-) Net has a large hole in it, (-) You are in downtown LA, (-) There are no fish in the water*

Table 2: Example of the three tasks in *PaCo*.

4 Tasks

Given the data collected in Section 3, we devise three complementary tasks to showcase the possible ways one could use the *PaCo* data to evaluate the current SOTA models' understanding of circumstantial preconditions. We select **P**reconditions **N**atural **L**anguage **I**nference (P-NLI) and **P**reconditions **M**ultiple-**C**hoice **Q**uestion **A**nswering (P-MCQA) as representative *discriminative* tasks, and **P**reconditions **G**eneration (P-G) task as a *generative* task. Table 2 summarizes the tasks and provides an example for each of them. In the rest of this section, we describe each task in detail and discuss the steps to prepare it from the raw precondition data. This preparation is fully automatic, and no human annotation or supervision signals have been used.

P-NLI Task Natural Language Inference (NLI) refers to tasks where given a sentence pair composed of a *hypothesis* and a *premise*, the system has to decide whether the hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given the premise (Williams et al., 2018). Each of the preconditions (e.g., "water is clean" or "water is polluted") of a statement can directly serve as a *premise* in the sense of NLI. Enabling preconditions correspond to *entailment* cases (e.g., "water is clean" *entails* "water is used for drinking"), whereas disabling preconditions can be annotated as *contradictions* (e.g. "water is polluted" *contradicts* "water is used for drinking"). The P-NLI task consists of 12.4K entries, with 6.6K entailment and 5.8K contradiction cases.

P-MCQA Task *PaCo* can also be directly converted to a multiple-choice question answering (MCQA) task in three steps. First, for each statement, each enabling (disabling) response is paired with three disabling (enabling) responses from the same statement. These three responses naturally act as negative samples (distractors), allowing us to have high-quality and fair questions. The question of the MCQA instance is then formed by appending "What makes this possible?" or "... impossible?" to the lexicalized statement. Second, in order to have more distractors and increase the number of multiple-choice instances we applied the two negative sampling methods used by Zhang et al. (2020b): Cosine Similarity Filtering, and Question/Answer Shuffling. Finally, in order to remove the annotation artifacts from the data, hence trivial instances, and prevent the models to exploit these artifacts instead of answering the questions, we used the *Lite* variation of the Adversarial Filtering method, which has been introduced in Sakaguchi et al. (2020) and formalized in Bras et al. (2020). This resulted in a P-MCQA task with 47K multiple choice questions, each with 4 choices.

P-G Task Despite our adversarial strategies, it remains possible that reasoning systems may identify annotation artifacts (Gururangan et al., 2018) in the data and solve the discriminative tasks without correctly performing the logical inference, as a result of those artifacts (Bras et al., 2020). Hence, we provide a third formulation as a generative commonsense reasoning task. In this task, we present the system with the exact question that has been presented to the human annotators, thereby mimicking the human annotation task of writing down the precondition as a natural language sentence. We then evaluate the model's response using the human responses as references. After removing the low-quality and *Invalid* responses from *PaCo*, the P-G task consists of 5.2K instances, with an average of 2.4 reference sentences per instance.

5 Experiments

This section pitches SOTA language models against the three tasks derived from *PaCo* (Section 5.1), dives deep into the tuning process to pinpoint time of comprehension (Section 5.2), investigates how LMs react to different relation types (Section 5.3), and finally revisits the distinction between soft and hard preconditions (Section 5.4).

5.1 Evaluating SOTA on *PaCo* Tasks

We assess our benchmark through evaluating representative NLP systems on the three tasks. This part starts with details about experimental setups (Section 5.1.1), followed by result analysis for the three tasks (Sections 5.1.3).

5.1.1 Experimental Setup

For each task, we start from available pretrained models and evaluate their performance on the test set in zero-shot and fine-tuned setups. To create the test set, we use a uniform random split of the statements that each task's instance is stemed from. For the split we use the $[0.45, 0.15, 0.40]$ ratio of the data for train/dev/test. The rationale for splitting based on the statements instead of the task instances is to prevent data leakage into the test sets through shared edges. The experiments are conducted on a commodity workstation with an Intel Xeon Gold 5217 CPU and an NVIDIA RTX 8000 GPU . For all the tasks, we use *allennlp* (Gardner et al., 2018) library for the TE model (Parikh et al., 2016) and use *huggingface* (Wolf et al., 2020) for the rest of them.

For the human evaluations of P-NLI and P-MCQA, we used a small (20) sample from test subset of each task and asked an expert to answer them. We then report the respective evaluation metric based on the task, as detailed below.

5.1.2 Evaluation Protocols

For P-NLI, we use *F1-Macro* score on the ground-truth labels and report the results on the unseen test split of the data.

For P-MCQA, we evaluate the systems' performance based on their default evaluation protocols as discussed below. For RoBERTa (Liu et al.,

Model	0-Shot	Tuned
AllenNLP TE	0.34	0.85
RoBERTa-large-MNLI	0.47	0.90
BART-large-MNLI	0.48	0.90
DeBERTa-base-MNLI	0.37	0.91
DeBERTa-large-MNLI	0.36	0.94
DeBERTa-xl-MNLI	0.37	0.91
Expert Human	1.0	-

Table 3: F1-Macro results of SOTA systems on P-NLI task based on *PaCo*

Model	0-Shot	Tuned
RoBERTa-base	0.23	0.34
RoBERTa-large	0.23	0.25
UnifiedQA-small	0.32	0.46
UnifiedQA-base	0.22	0.56
UnifiedQA-large	0.23	0.63
Expert Human	0.95	-

Table 4: Accuracy results of SOTA systems on P-MCQA task based on *PaCo*

2019), we use the LM coupled with a linear regression layer as classification head. In this method, the LM is tasked with embedding each question/answer pair, and the classification head assigns a score to the pair. Later for each MC instance, the question/answer pair with the highest score is selected as the output choice. We report the accuracy score (code from (Pedregosa et al., 2011)) based on the output choices from the model. For UnifiedQA, we follow the original setting by Khashabi et al. (2020) to let the model conduct sequence-to-sequence generation based on the question. Here, the question and all choices are feed to the model, and it is expected to generate the correct choice's text. We then report the f1 score by selecting the one that is closest to the generated answer from the candidate choices.

For P-G, to automatically evaluate the machine-generated answers of the models, we use *Bleu-2* (Papineni et al., 2002) (code from (Bird et al., 2009)) and *ROUGE-2* (Lin, 2004) (code from (Wolf et al., 2020)) metrics. We do not use methods with large n-gram match (e.g., *Bleu-4*) for two reasons. *First*, the small number of reference sentences (at most 3) made most of model's output not matching any reference sentence. *Second*, relatively short reference sentences leads to no 4-gram match and mostly zero *Bleu-4* scores.

For the human evaluation score of the machine generated responses, we sample 100 responses and use a method similar to *quality control* method in Section 3.2 (here we consider the *Truism* responses as *Informative*), and report the percentage of *informative* responses from tuned models.

5.1.3 Results and Discussions

We hereby separately discuss the performance of SOTA models on the three tasks in details.

(1) P-NLI Results As shown in Table 3, all systems tend to get near-random results in the zero-shot setup. In case of the *BART-large-MNLI* model, although the zero-shot *F1-Macro* score is higher, it is far from human-level score (1.00). We observe that even models that are trained on large and diverse learning resources (e.g. MNLI (Williams et al., 2018)) are not able to perform well on the P-NLI in a zero-shot fashion.

This high scores after fine-tuning can be attributed to systems' exploiting the annotation artifacts of data instead of learning to reason with preconditions. This claim will be further supported by the P-MCQA results.

(2) P-MCQA Results The P-MCQA has all the intricacies of the original precondition data absent from the simple annotation artifacts that make it a better alternative to evaluate systems. As presented in Table 4, there is a significant gap between the ideal and machine performance in the P-MCQA benchmark that further supports the novelty of *PaCo* and tasks stemming from it.

After investigating the answers, we observe that even the promising large models tend to confuse the enabling v.s. disabling cases. For example the *UnifiedQA-Large* model, mistakenly chooses a disabling response "Your car is out of fuel" for the enabling question "Gas are typically used for providing energy. What makes this possible?". This might be explained by the statement that LMs tend to focus more on correlation of lexical occurrences and statistical patterns (e.g., gas and car/fuel), rather than the actual question. In addition, similar to Zhou et al. (2020), we observe that LMs lack understanding of linguistic permutations like negations, and lean toward positive words.

(3) P-G Results As summarized in Table 5, the automatic evaluation results, BLEU and ROUGE, are close to zero for all models. This shows that the models fall short in generating similar to reference precondition even after fine-tuning. On the other hand, the human annotation sheds more light on the results and show the relative comparison of the models.

Here the automatic evaluation methods do not sufficiently distinguish between the models as the difference among them are negligible. Hence,

Model	BLEU		ROUGE	HUM
	0-Shot	Tuned	Tuned	Info.
UnifiedQA-small	0.007	0.157	0.064	0.12
UnifiedQA-base	0.006	0.303	0.115	0.28
UnifiedQA-large	0.029	0.330	0.128	0.48
BART-base	0.046	0.091	0.140	0.19
BART-large	0.041	0.058	0.117	0.11
GPT2	0.097	0.133	0.067	0.36
Expert Human	-	-	-	1.0

Table 5: BLEU-2, ROUGE-2, and human evaluation Information score for results of SOTA systems on the P-G task. Zero-shot ROUGE scores are omitted due to lack of information.

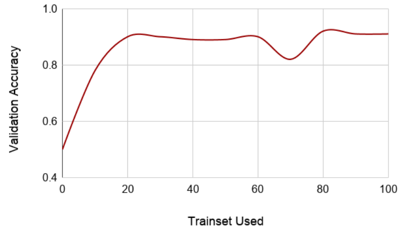

Figure 2: F1-Score of fine-tuning *RoBERTa-large-MNLI* with increasing amounts of training data from P-NLI.

the comparison rather provides complementary insights to the two discriminative tasks. This is consistent with similar generation tasks (Rudinger et al., 2020), due to the small number of reference responses and relatively large space of correct responses that makes automatic evaluation of such machine responses an unresolved problem (Chen et al., 2020).

Upon analyzing the results we noticed several patterns in the generated responses. First, models tend to generate simple answers mostly discussing the existence or availability of the subject. For example, *BART-base* frequently generated patterns such as " <head> is closed" or "You have <head>" some of which were informative. Second, similar to the P-MCQA task, the models tend to confuse enabling and disabling preconditions. For example, *BART-large* generated the enabling precondition "The clothes are dirty" instead of disabling precondition for the statement "Washing clothes are used for making fresh again".

5.2 Diving in the Tuning Process

In the above evaluation on P-NLI, we observe that all models get higher scores after fine-tuning. Here, we investigate the fine-tuning process to find at what point the model understands the requirements of the task.

Experimental Setup We focus on the *RoBERTa-large-MNLI* (Liu et al., 2019) model in the P-NLI task. The experimental setup is similar to section 5.1.1. We evaluate the model's performance on the test split of P-NLI in checkpoints during the tuning process instead of just at the end of it. Checkpoints are based on the amount of tuning data the model has observed ($10\%, 20\%, \cdots, 100\%$).

Results Figure 2 plots the changes of score of the model as it gets more tuning data. The slow saturation of the F1 score here suggests that the instances in P-NLI are not trivial for the model and it actually has to see a lot of instances to be able to perform the task. Considering that the *RoBERTa-large-MNLI* has been pre-trained on a vast corpus, our result shows the novelty and uniqueness of the *PaCo* data.

5.3 Discussion on Different Relation Types

Given that *PaCo* consists of three relations types, we next pose the question of how well the LMs can handle each relation type. Here, we break down the results presented in Section 5.1 per relation type and discuss the model performance on each type.

Experimental Setup Due to simplicity of automatic evaluation, we on focus on the two discriminative tasks, P-NLI and P-MCQA. The experimental setup here is similar to section 5.1.1, except that for both zero-shot and fine-tuned settings where we measure the dissected results based on the relation types as well as their aggregation.

Results On the P-NLI task, similar to the challenges for human annotators (Section 3.2), all NLI models tend to get lower accuracy on instances derived from *Causes* and *Desires* relations, compared to *Usedfor*. For instance, the *DeBERTa-large-MNLI*, has a 6% gap between the performance on *UsedFor* and *Causes* instances. In the P-MCQA task, we observe a similar pattern between *Causes* and *Desires* relations on one hand, and *Usedfor* on the other hand. For instance, the *UnifiedQA-large* mode shows a 13% gap between instances with *Usedfor* and *Desires* relations. The detailed P-NLI and P-MCQA performance results dissected based on relation types are provided in Tables 9 and 10 in the Appendix section.

5.4 Hard and Soft Preconditions

In this work, we argued for the use of hard preconditions as opposed to soft preconditions used in previous works. Although semantically different, one may argue that using soft preconditions may help the models learn the task of reasoning with preconditions with already existing data. In this section we test this hypothesis.

Experimental Setup Using the approach presented in Section 4, we created an NLI resource from two available resources with soft preconditions: Rudinger et al. (2020) and ATOMIC2020 (Hwang et al., 2020) (Details in Appendix B.3). We focused on the *RoBERTa-large-MNLI* (Liu et al., 2019) model, fine-tuned in on the two resources, and evaluate on the test set of P-NLI. The experimental setup here is similar to Section 5.1.1.

Results Although these resources have an order of magnitude more data (88K instances in ATOMIC2020 (Hwang et al., 2020) and 236K instances in Rudinger et al. (2020)), there is more than 10% gap between the performance of the model tuned on them in the P-NLI task compared to a model exposed to *PaCo* data. Table 11, presents the detailed results of tuning *RoBERTa-large-MNLI* model on each of the NLI-style datasets, while being evaluated on P-NLI's test subset.

6 Related Work

Resources of Preconditions. A few resources have provided representations for preconditions of statements. ConceptNet (Speer et al., 2017)'s *HasPrerequisite* relation, ATOMIC (Sap et al., 2019a)'s *xNeed* relation, and CauseNet (Heindorf et al., 2020) data can express concept dependencies, such as, e.g., before one bakes bread, they need to buy ingredients and go to a store. Instead of adding new edges, our work annotates existing edges with contextual preconditions, which helps reasoners understand when to use an edge and when not to. ASER (Zhang et al., 2020a) and ASCENT (Nguyen et al., 2021) extract edges from unstructured text together with their associated context. As such, their knowledge is restricted by information available in text, and they do not express *disabling* preconditions. It is also unclear to which extent their contextual edges express *enabling* preconditions, rather than coincidental information. GLUCOSE (Mostafazadeh et al., 2020) comes closer to our work, as they also extract *enabling* preconditions (e.g., *Possession state that enables X*) via crowdsourcing. Similarly, PeKo (Kwon et al., 2020) extract *enabling* preconditions between event pairs from available text and use it to propose precondition identification and generation tasks between pair of sentences. However focusing only on causal relations in available text hinders the extent of their tasks. Both GLUCOSE and PeKo do not explore disabling preconditions.

Reasoning with Preconditions. Few efforts have been made on evaluating commonsense reasoning with preconditions. Rudinger et al. (2020) focus on modeling weakeners and strengtheners of commonsense statements. Their work adds a *utility* sentence to the *hypothesis-premise* pair in NLI-style tasks and ask whether it weakens or strengthens the relationship of the pair. Similarly, Hwang et al. (2020)'s *Hindered by* and *Causes* also focuses on similar relationship for events with focus on presenting a knowledge resource.

Our work differs as we focus on a crisp condition of *enabling/disabling* that can be particularly useful in logic-like reasoning tasks (as opposed to probabilistic inference). In addition, our task allows the reasoning to be processed as canonical NLI and can benefit from existing NLI architectures instead of modifying them.

7 Conclusions and Future Work

We presented, *PaCo*, a dataset of 12.4K collected enabling and disabling preconditions of everyday commonsense statements from ConceptNet. We utilize this resource to create three tasks for evaluating the ability of systems to reason over circumstantial preconditions, namely: P-NLI, P-MCQA, and P-G. Our evaluation shows that SOTA reasoners largely fall behind human performance, indicating the need for further investigation to develop precondition-aware systems.

Future work should cover the inclusion of preconditions in logical reasoning of the neuro-symbolic reasoners. It should also expand to multi-modal setup or investigate using weak-supervision to gather preconditions. Alternatively, we can leverage the contributed resource to develop generative models for automated context-aware knowledge base construction (Sorokin and Gurevych, 2017).

Ethical Statement

Though we may present this as we started from openly available data that is both crowdsource-contributed and neutralized, however it still may reflect human biases (Mehrabi et al., 2021).

During our data collection we did not collect any sensitive information, such as demographic or identity characteristics. We only limited the annotators to English-speaking users from mainly English-speaking countries such as US, which may add cultural bias to the data. However, neither our crowd annotators or the expert annotators noticed any offensive language in the questions or the responses.

Given the urgency of addressing climate change we have reported the detailed model sizes and runtime associated with all the experiments in Appendix C.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7456–7463. AAAI Press.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Yohan Chalier, Simon Razniewski, and Gerhard Weikum. 2020. Joint reasoning for multi-faceted commonsense knowledge. *arXiv preprint arXiv:2001.04170*.

Anthony Chemero. 2003. An outline of a theory of affordances. *Ecological psychology*, 15(2):181–195.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6521–6532, Online. Association for Computational Linguistics.

Kevin Crowston. 2012. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Richard E Fikes and Nils J Nilsson. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Eleanor J Gibson. 2000. Where is the information for affordances? *Ecological Psychology*, 12(1):53–56.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Stefan Heindorf, Yan Scholten, Henning Wachsmuth, Axel-Cyrille Ngonga Ngomo, and Martin Potthast. 2020. Causenet: Towards a causality graph extracted from the web. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 3023–3030. ACM.

Jerry R Hobbs. 2005. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209.

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*.

Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L McGuinness, and Pedro Szekely. 2021. Dimensions of commonsense knowledge. *arXiv preprint arXiv:2101.04640*.

Filip Ilievski, Pedro Szekely, Jingwei Cheng, Fu Zhang, and Ehsan Qasemi. 2020a. Consolidating commonsense knowledge. *arXiv preprint arXiv:2006.06114*.

Filip Ilievski, Pedro Szekely, and Daniel Schwabe. 2020b. Commonsense knowledge in wikidata. In *ISWC Wikidata workshop*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Heeyoung Kwon, Mahnaz Koupaee, Pratyush Singh, Gargi Sawhney, Anmol Shukla, Keerthi Kumar Kallur, Nathanael Chambers, and Niranjan Balasubramanian. 2020. Modeling preconditions in text with a crowd-sourced dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3818–3828, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China. 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*.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 22–32, Hong Kong, China. Association for Computational Linguistics.

Ninareh Mehrabi, Pei Zhou, Fred Morstatter, Jay Pujara, Xiang Ren, and Aram Galstyan. 2021. Lawyers are dishonest? quantifying representational harms in commonsense knowledge resources. *arXiv preprint arXiv:2103.11320*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2021. Advanced semantics for commonsense knowledge extraction. *The Web Conference (WWW 2021)*.

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

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 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*.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3027–3035. AAAI Press.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Yoav Shoham. 1990. Nonmonotonic reasoning and causation. *Cognitive Science*, 14(2):213–252.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789, Copenhagen, Denmark. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Adina Williams, Nikita Nangia, and Samuel 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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, Online. Association for Computational Linguistics.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4970–4977. AAAI Press.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020a. ASER: A large-scale eventuality knowledge graph. In *WWW '20:*

The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pages 201–211. ACM / IW3C2.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

Pei Zhou, Rahul Khanna, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2020. Rica: Evaluating robust inference capabilities based on commonsense axioms. *EMNLP-Findings*.

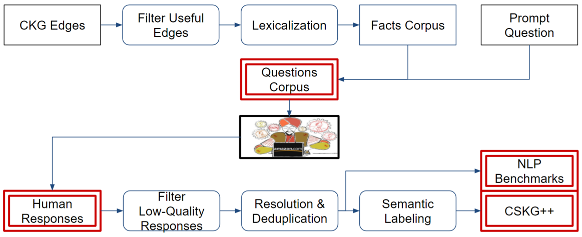

Figure 3: Data-collection and processing in a nutshell

Figure 4: A sample question-unit used in main survey on the AMT

A Data Collection Details

We used Amazon Mechanical Turk (AMT) (Crowston, 2012) to collect the *PaCo*. This enabled us to coordinate the study and access a large pool of English-speaking participants as our study population. The AMT is especially suitable for this study as it can facilitate accessing a diverse population of participants which is necessary for any notion of commonsense. Our study on AMT consists of two parts: a tutorial that also serves as a qualification test and the main survey. In addition, we implemented two levels of quality control: in the first one we use a response checker code and in the second we use human annotators to ensure only high-quality responses wind up into the final data.

A.1 Main AMT Survey

In the main survey, the participants are given a set of question-units (sample in Fig. 4) each consists of a factual sentence (discussed in Section A.2) followed by a prompt question, then we ask participants to write their responses for each prompt question in the designated text box in front of the unit. The prompt questions are short questions that ask about the preconditions that enable or disable the factual sentence (e.g. *what makes this possible?*, *when is this impossible*). The goal of this phase is to use the powers of crowdsourcing to capture as much information as needed to create a dataset of enabling and disabling conditions.

A.2 Gathering Factual Sentences

The first row in Fig. 3 summarizes the steps to create the factual sentences. Each factual sentence is a short sentence derived from an edge from a commonsense knowledge graph. The information on this knowledge graph is related to everyday situations such as usage of objects (*A net is used for catching fish.*), or capabilities of objects (*Humans are capable of catching a bus.*), etc. (Speer et al., 2017; Ilievski et al., 2020a; Sap et al., 2019a). In our case, the knowledge associated with each factual sentence is extracted from ConceptNet (Speer et al., 2017), a well known commonsense resource. To limit the scope of this work we only focus on *UsedFor*, *Causes*, and *Desires* relations from ConceptNet, however, the method can be extended to any other relation from any other knowledge graph.

To convert the knowledge graph edges to human-readable factual sentences, we used automatic lexicalization methods, similar to (Ma et al., 2019; Bouraoui et al., 2020). In this method, we define a set of templates to convert the edge to a set of sentence candidates, then use the perplexity score of a language model to pick the best candidate for each edge. The lexicalization is explained in more details in Appendix A.4.

Since ConceptNet's knowledge is not perfect, some of the generated factual sentences may not fully make sense. Additionally, the automatic conversion of edges to the sentence is not perfect, hence some sentences may have odd grammar (e.g. *An net is used for catch fish*). Consequently, some of the question-units may be hard to understand or just be wrong. To help us find those question-units and ignore them in future iterations, each unit is presented with an adjacent checkbox labeled *This does not make sense*. The participant may choose to select the checkbox and skip answering that prompt. To make the payment structure fair for the participants, they will get paid regardless of their response.

A.3 Qualifying Participants

To ensure the participants can understand the task, we prepared detailed instructions that explain to the participants what they need to do and what are the criteria for a good vs bad response. For example, in the instructions, we ask participants to avoid using negative sentences or avoid using pronouns to refer to objects. The instruction is 366 words with an expected reading time of < 5 mins. Additionally, we have prepared a set of good/bad examples associated with each rule that can also be accessed in the tutorial. Each one of the good/bad examples comes with a short explanation clarifying the reason for its good/bad rating.

The participants are then asked to take the qualification test as a check on whether they have read and understood the instructions. The qualification test contains 10 multi-choice questions (each with two choices); each containing a question-unit (similar to those that are used in the main survey) with two choices of the possible responses that one may give to them. We have carefully designed each multiple-choice question such that it tests the participants' understanding of the rules individually and give them feedback on their wrong answers. For example, for the rule discouraging the use of negative sentences, we have two questions where the wrong answers contain a negative verb. After successfully passing the test, participants with acceptable scores are granted a qualification badge that allows them to engage in the main survey. It must be noted that the detailed instructions and the good/bad examples are both available in the main survey as a memory refresher for the participants.

A.4 Edge Lexicalization

Each of the selected edges is lexicalized using a combination of templates and masked LMs described by Ma et al. (2019) and Bouraoui et al. (2020). Similar to Ma et al. (2019), we use a combination of the templates for each relation (e.g. *[subject] is used for [object]*, *[subject] is used by [object]*) and use the perplexity score from the LM to select the best lexicalization for each edge. However, this method does not guarantee the selection of the best lexicalization as the perplexity score reflects the probability of the sentence tokens appearing in that specific order rather than the sentence's grammatical correctness. To mitigate this issue, in addition to the above method, following (Bouraoui et al., 2020), we let the LM adjust the templates as well by adding one masked token to some templates (e.g. *[subject] is used [MASK] [object]*) and let the LM fill the *mask* before filling the *subject* and the *object* slots of the template.

B Results in More Details

B.1 Edge Selection Results

In this section, we provide further evidence to support the decision to use the *UsedFor* edges without any additional filtering. First, we showcase the lack of correlation between a hand-annotated usefulness indication of the precondition statements and existing quantitative methods/scores. Then, in a similar setup, we show that the *UsedFor* edges have a higher usefulness score.

Metric	[0,10](%)	[50,60](%)	[90,100](%)
Perp.	75	95	90
Salient	80	100	95
Weight	95	90	90

Table 6: hand-annotated usefulness indication of the precondition statements for top/bottom/mid percentile buckets of the quantitative methods. The $[A, B]$ label indicates edges with the metric score in the range of $[A, B]$ percentile of the metric score.

Metric	Score(%)
UsedFor	95
CapableOf	90
RelatedTo	40

Table 7: hand-annotated usefulness indication of the precondition statements three of the ConceptNet relations

For the first study, we only focus on *UsedFor* edges. For each metric, we randomly sample 20 edges in each percentile of the metric and hand-annotate the usefulness of sampled edges in each percentile. Then, for each percentile-metric, we report the percentage of edges that were considered useful for our study. The results in Table 6, summarizes the usefulness score for three of the percentile buckets for three of the metrics. For the *perplexity* score we used the RoBERTa (Liu et al., 2019) language model on the lexicalized edges, for the *Salient* score we used DICE metrics (Chalier et al., 2020), and for the *weight* score we use the weights from the ConceptNet (Speer et al., 2017) itself. The usefulness scores suggest that a higher score may or may not result in more useful edges which makes using them for filtering edges tricky. This study is by no means conclusive due to both the small sample sizes and a small number of trials, however, it led us to choose the edges solely based on relation type and leave further filterings to future work.

For the second study, Table 7, we group edges based on their relations only and compute the usefulness score for each relation. The results showed that *UsedFor* edges tend to generally be more useful for our annotation task. This couple with the statement that *UsedFor* edges could be annotated with both enabling and disabling preconditions led us to focus on them for this study.

B.2 Additional Results from P-NLI

Table 8 presents some error cases that each model predicts on the test subset of P-NLI.

As our version of NLI only consists of *Entailment* and *Contradiction* labels, we discuss the results using binary classification terminology.

In addition, the detailed results of Table 3 dissected by the relation types are provided in Table 9.

B.3 Details of Soft Preconditions on P-NLI

In order to convert the ATOMIC2020 (Hwang et al., 2020) to an NLI-style task, we method similar to P-NLI and focused on three relations *HinderedBy*, *Causes*, and *xNeed*. From these relations, *HinderedBy* is converted to *Contradiction* and the rest are converted to *Entailment* instances.

For converting Rudinger et al. (2020), we focused on SNLI subset of their data and used the concatenation of SNLI's "Hypothesis" and "Premise" as hypothesis and their "Update" sentence as premise.

Table 11, presents the detailed results of tuning *RoBERTa-large-MNLI* model on each of the NLI-style datasets, while being evaluated on P-NLI's test subset.

C Model Sizes and Run-times

For table 3, Runtimes: TE=2hr,rbrta=2.5hr, dbrta-base=0.5hr, dbrta-large=2hr, dbrta-xlarge=3.5hr, BART-large=2hr and #params: TE=0.5M, rbta=356M, dbrta-base=141M, dbrta-large=401M, dbrta-xlarge=751M, BART-large=407M. For table 4, Runtimes:rbta-base=1hr, rbta-large=2hr, uqa-small=1hr, uqa-base=4hr, uqa-large=20hr and #params: rbta-base=124M,rbta-large=355M, uqa-small=60M, uqa-base=222 M,uqa-large=737M. In table 1, Runtimes: uqa, gpt2=10min and #params: gpt2=1.5B. Finally in table 5, Runtimes:uqa-small=1hr, uqa-base=2hr, uqa-large=6hr, gpt2=1.5B, bart-base=139M, bart-large= and #params: uqa-small=60M,uqa-base=222 M, uqa-large=737M, gpt2=1.5B, bart-base=139M, bart-large=406M.

Model	*Statement*	*Context*	*
TE	You can typically use self adhesive label for labelling things	The self adhesive label runs out of glue.	FP
	Acoustic ceiling is typically used for dampening sound.	in rooms with noise above a certain decibel.	FP
	You can typically use self adhesive label for labelling things.	Labeling things that are wet.	FP
	Farm is typically used for raising crops.	Enough rain should be available.	FN
roberta	You can typically use pets to provide companionship	the pet is dog.	FN
	Acoustic ceiling is typically used for dampening sound	The sound is too loud	FP

Table 8: Test results of SOTA systems on NLI task based on the *PaCo*. FP: False Positive, FN: False Negative

Model	Rel.	0-Shot	Tuned
RoBERTa-large-MNLI	UsedFor	0.34	0.85
	Causes	0.48	0.90
	Desires	0.48	0.90
	All	0.47	0.90
BART-large-MNLI	UsedFor	0.51	0.91
	Causes	0.41	0.82
	Desires	0.46	0.89
	All	0.48	0.89
DeBERTa-base-MNLI	UsedFor	0.37	0.91
	Causes	0.32	0.84
	Desires	0.38	0.88
	All	0.37	0.89
DeBERTa-large-MNLI	UsedFor	0.38	0.94
	Causes	0.31	0.88
	Desires	0.36	0.90
	All	0.36	0.92
DeBERTa-xlarge-MNLI	UsedFor	0.37	0.94
	Causes	0.31	0.88
	Desires	0.37	0.89
	All	0.37	0.91

Table 9: F1-Macro results of SOTA systems on P-NLI task based on *PaCo* dissected based on relation type

Model	Rel.	0-Shot	Tuned
RoBERTa-base	UsedFor	0.24	0.39
	Causes	0.25	0.43
	Desires	0.22	0.27
	All	0.23	0.34
RoBERTa-large	UsedFor	0.25	0.23
	Causes	0.20	0.29
	Desires	0.22	0.25
	All	0.23	0.25
UnifiedQA-small	UsedFor	0.35	0.52
	Causes	0.35	0.45
	Desires	0.27	0.40
	All	0.32	0.46
UnifiedQA-base	UsedFor	0.23	0.64
	Causes	0.17	0.54
	Desires	0.22	0.49
	All	0.22	0.56
UnifiedQA-large	UsedFor	0.27	0.70
	Causes	0.20	0.64
	Desires	0.20	0.57
	All	0.23	0.63

Table 10: Accuracy results of SOTA systems on P-MCQA task based on *PaCo*

Tune Dataset	Relation	F1-Macro
PaCo	UsedFor	0.85
	Causes	0.90
	Desires	0.90
	All	0.90
Hwang et al. (2020)	UsedFor	0.50
	Causes	0.50
	Desires	0.45
	All	0.48
Rudinger et al. (2020)	UsedFor	0.84
	Causes	0.80
	Desires	0.82
	All	0.83

Table 11: Results of RoBERTa-large-MNLI model on test set of P-NLI after being tuned on different datasets, dissected based on relation type.