# Ontology Knowledge-enhanced In-Context Learning for Action-Effect Prediction

**Fangjun Li**[1] SCFLI@LEEDS.AC.UK
**David C. Hogg**[1,3] D.C.HOGG@LEEDS.AC.UK
**Anthony G. Cohn**[1,2,3,4] A.G.COHN@LEEDS.AC.UK
[1] School of Computing, University of Leeds, Leeds, LS2 9JT, UK
[2] College of Electronic and Information Engineering, Tongji University
[3] Alan Turing Institute, UK
[4] School of Control Science and Engineering, Shandong University

## Abstract

Implementing human-level reasoning about action effects is an important competence for a cognitive agent: given precondition and action descriptions, a system should be able to infer the change in the physical world that the action causes. In this work, we propose a new action-effect prediction task. We explore few-shot learning with large pre-trained language models based on a limited number of samples and propose task-relevant ontology knowledge (from KnowRob ontology) integration for in-context learning with generative pre-trained transformer (GPT) models. Specifically, we develop an ontology-to-text transformation to bridge the gap between symbolic knowledge and text. We further introduce unseen knowledge learning via GPT-3 to infer knowledge for concepts that do not have definitions in the knowledge base. We evaluate our proposed method on two human-annotated datasets. Experimental results demonstrate that our approach can improve the performance of large-scale, state-of-the-art models on two action-effect prediction datasets.

## 1. Introduction

The ability to predict the outcome of an action and the consequent world state is crucial for cognitive agents to successfully plan and perform complex tasks (Alomari et al., 2022). When a cognitive agent performs navigation or motion control, it needs act to achieve its goals. Once an action is performed during the execution of a task, the world state changes, affecting the agent's possibilities for completing the task. To successfully execute tasks, the agent should plan the goal in accordance with such changing world status (Cox et al., 2017). Therefore, action-effect prediction is an important component of cognitive system goal planning. We define our action-effect prediction task as follows: given the textual description of an action, a sentence describing the subsequent world state after the action should to be generated. For example, given an action description 'cutting a cucumber', then the output effect description might be 'the cucumber no longer exists but is now a collection of slices'. Our approach to this task is to use a language model, such as GPT-3, exploiting a prompt (a piece of text in natural language with the description of the task for the pre-trained

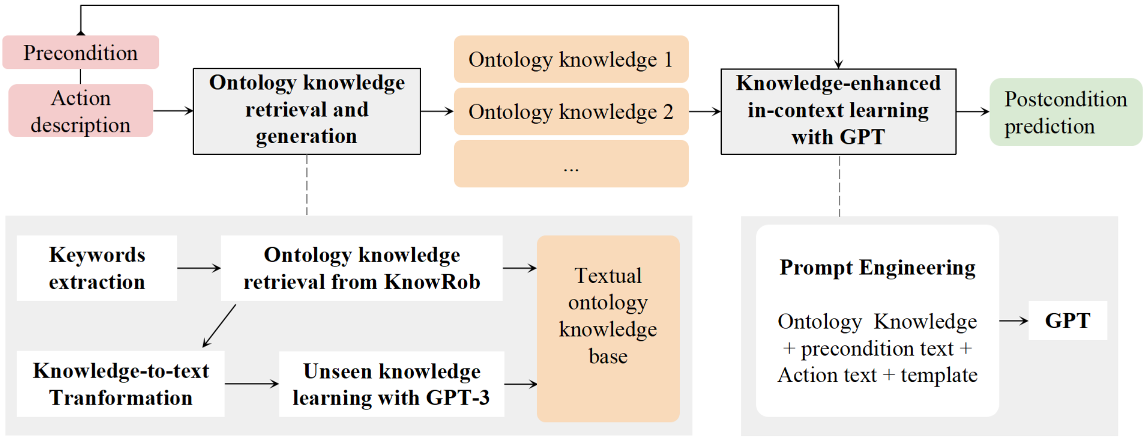

Precondition
Action description
Ontology knowledge retrieval and generation
Ontology knowledge 1
Ontology knowledge 2
...
Knowledge-enhanced in-context learning with GPT
Postcondition prediction
Keywords extraction
Ontology knowledge retrieval from KnowRob
Textual ontology knowledge base
Knowledge-to-text Tranformation
Unseen knowledge learning with GPT-3
Prompt Engineering
Ontology Knowledge + precondition text + Action text + template
GPT

*Figure 1.* The pipeline of our knowledge-enhanced in-context learning for action-effect prediction. We form a knowledge retrieval and generation process to get ontology knowledge from the KnowRob base. Through ontology knowledge-enhanced prompt engineering, the textual knowledge pieces, together with precondition and action descriptions, are employed as input to the GPT model to do post-condition action-effect prediction. The arrows represent the data flow. The dashed lines connect boxes in the upper part of the figure to a more detailed explanation of that system component. (Note that the notion of "Prompt Engineering" is explained in section 2.1.)

language model to interpret and complete) based on a commonsense ontology of action effects, as described further below.

For this generation task, the labelled data generally accessible includes some well known cases presented in previous studies (*e.g.*, Davis (1998)'s egg cracking problem), manually collected effect annotations for 140 actions collected by Gao et al. (2018), and 2000 interactions in terms of initial world state, the action, and subsequent world state after the action (Zellers et al., 2021). Data from various sources vary in quantity and level of detail. The description of an action can range from a verb-noun phrase to a paragraph describing a process in which the action is only a component. **How to utilise these diverse, small-scale datasets is the primary challenge.** Rather than building a model based on large-scale generative language models (LMs) from scratch for a specific dataset (Zellers et al., 2021), **we look into the few-shot learning capabilities of pre-trained language models (PLMs) that could be easily extended to different datasets in this task.** The third iteration of the generative pre-trained transformer (GPT-3) is capable of performing few-shot in-context learning, where an LM "learns" to do a task simply by conditioning on a text input describing a new task with input-output examples. This eliminates the need to do task-specific fine-tuning on thousands or tens of thousands of examples. For our task, we build input text (also known as the prompt) based on the precondition and action descriptions to obtain primary post-condition predictions from GPT-series models. The text prompt for each action query consists of two parts: some instances of action description contexts and action-effect completions, and one final action context to query the model.

Even though the GPT-series models, particularly GPT-3, are effective at text generation, **there is still a long way to go before reaching robust human-level reasoning** (Zhou et al., 2020). **We want to make GPT-3 a better predictor on the textual action-effect inference task by incorporating knowledge from a task-related knowledge base.** We build a knowledge-enhanced in-context learning approach for GPT-3 on our action-effect prediction task. Currently the majority of works fusing knowledge into LMs for commonsense reasoning tasks (Shwartz et al., 2020; Liu et al., 2021b; Bian et al., 2021) consider Knowledge Bases (KBs) that cover a broad range of topics, among which Wikidata (Vrandečić & Krötzsch, 2014) and ConceptNet (Speer et al., 2017) are the most commonly used knowledge resources. We hypothesise that task-related knowledge could help LMs solve tasks. Specifically, for our action-effect task, we employ KnowRob (Tenorth & Beetz, 2013) which supplemented OpenCyc with additional knowledge about human everyday activities and household objects, and thus matches our requirements for use in the action effect inference task. The "Knowledge-enhanced in-context learning" part of Fig. 1 shows how we make use of such ontology knowledge: the ontology knowledge together with precondition and action descriptions, form the knowledge-enhanced prompt feeding to the GPT model.

KnowRob as a fixed ontology knowledge base has the knowledge incompleteness problem: there are no explanations for certain terms. **How to obtain knowledge for concepts that do not have definitions in the fixed knowledge base and use them in LMs is a major challenge.** The "Ontology knowledge retrieval and creation" part of Fig. 1 depicts our knowledge processing methods. First, We retrieve the logical representations for action-relevant concepts (verbs and nouns in action phrases) from KnowRob. Then, the retrieved pieces of knowledge are converted into English text. We concatenate these terms and transformed knowledge pieces to form a *prompt*, which will be used as input for GPT-3 to infer new knowledge. For concepts without definitions in KnowRob, we query GPT-3 with the prompt and utilise the returned text as knowledge for those terms. In this way, we can get knowledge for all terms.

We conducted experiments on the second and third generation of GPT models: GPT-2 and GPT-3. GPT-3 is not publicly accessible: access can only be obtained through an API, with users required to pay for token usage. Unlike GPT-3, the GPT-2 model can be accessed using Hugging Face's open-source Transformers library (Wolf et al., 2020). We compare the generation results with and without external ontology knowledge on two human-annotated action-effect prediction datasets. Experiments show that the prediction performance improves when exploiting completed ontology knowledge from KnowRob. We have released the code and results on Github[1].

## 2. Related Work

There are many reasoning tasks relating to action: e.g., commonsense inference on events (Rashkin et al., 2018), action planning (Fine-Morris et al., 2020), extracting action sequences from text (Olmo et al., 2021), tracking and answering questions about how entities change during actions (Tandon et al., 2018), reasoning about causal and temporal actions (Xiao et al., 2021). The closest task about action-effect is addressed by Tandon et al. (2018), though they formulate the action-effect prediction task as a multi-class classification problem. They characterize the effect of actions by state changes

[1] https://github.com/lfj95/action-effect-gpt

of each entity on four pre-given states. This brings a serious limitation: only broad entity states are included, which cannot accurately represent detailed state changes.

We argue that generation, rather than choice-based QA, is of more practical value. For example, for ‘heat water,’ the open-ended inference ‘water will begin to boil, as evidenced by bubbles forming and rising to the surface’ contains significantly more information than state ‘none’. Rashkin et al. (2018) proposed a generation-based commonsense inference task: given an event, the system reasons about the likely intents and reactions. They focus on events rather than actions. The work by Zellers et al. (2021) gathered human-annotated data detailing actions, pre- and post-conditions, paving the way for our investigation. However, the problem they solve differs from ours in that they use an object states list as input to their natural language generation task rather than precondition descriptions.

Furthermore, lots of previous action-relevant commonsense reasoning tasks utilising language models based on deep neural network architectures (rather than traditional KR attempts using hand built KBs) benefited from large (action-specific) training data: (Tandon et al., 2018) use the ProPara dataset (Dalvi et al., 2019), which contains a pre-given paragraph describing a sequence of actions about each given topic. NExTQA (Xiao et al., 2021) contains thousands of manually annotated question-answer pairs grouped into causal action reasoning questions. Rashkin et al. (2018) construct a crowd-sourced corpus of 25k event phrases encompassing a wide variety of everyday events and situations. It’s important to give thought to how to predict action effects with limited and varying data.

## 2.1 Few-shot Learning with PLMs

With limited labelled action-effect pairs as training examples, we aim to learn a language generation model in this data-scarce situation. The powerful GPT-3 model (Brown et al., 2020) can do few-shot learning given a concatenation of training examples, without parameter updates, on downstream tasks. That means working with a large fixed PLM and only the representation of the prompt is learned. PLMs have also shown promise in cognitive systems. For example, Wray et al. (2021) proposed language models be used as a source of knowledge for cognitive systems. Their subsequent work (Kirk et al., 2022) explores using responses from GPT-3 as a knowledge source for a mobile robot to perform novel tasks in one shot in a household environment.

Lu et al. (2022) shows that the performance of GPT-3’s in-context learning is highly sensitive to the input prompts: the prompt format, the training examples, and even how they are put together all have a big impact on how well it does. Then there is **prompt engineering**, which refers to the technique used in conjunction with large LMs such as GPT-3 whereby, when given an input ‘prompt’ , the LM will then output some further text in response. E.g., the prompt “In summary," aims to prompt the LM to summarise the preceding text. Different prompts may be more or less effective and *prompt engineering* refers to engineering of the most effective prompts for a particular envisaged task such as summarisation, or sentence completion. Liu et al. (2021a) gave a formal description of prompting and prompt engineering, explaining the basic steps for building a prompt function $f_{prompt}(x)$. One important step is to find an appropriate prompt shape for the textual *template*. Notwithstanding their in-depth review of prompting methods, the question of how best to incorporate knowledge into prompts remains unresolved.

### 2.2 Integrating External Knowledge into LMs

Prior to the availability of vast amounts of data for neural approaches, much work was devoted to symbolic reasoning through deduction. Many knowledge resources have been developed during the previous few decades, *e.g.*, (Lenat, 1995; Miller, 1995; Bosselut et al., 2019). Bosselut et al. (2019) indicated that effective utilisation of KBs may enhance the PLMs' capacity for commonsense reasoning, while Wu et al. (2022) suggested that access to external knowledge is essential for many natural language processing tasks, such as question answering and dialogue. Some work (Liu et al., 2021b; Bian et al., 2021; Kapanipathi et al., 2020) tried to improve a LMs' commonsense reasoning ability using a knowledge base, but the majority of these works consider knowledge bases that cover a broad range of topics, among which Wikidata and ConceptNet are the most commonly used knowledge resources.

Apart from the above mentioned KBs, there are some other KBs where a great deal of effort has been put in their development over the past few decades, but little effort currently considers utilising those knowledge sources to enhance LMs' capabilities. Symbolic ontology KBs are a representative example. They are created by domain experts, and are generally of high quality. But very limited work uses them in conjunction with LMs; this might be due to their symbolic formulations, which add complexity when integrating knowledge into LMs, or to the insufficient breadth of the knowledge source. Cyc (Lenat, 1995), which represent millions of common sense facts in a machine-readable format is by far one of the most cited ontology knowledge resources. It is not open source, but a smaller version called OpenCyc has been built and made available for free. OpenCyc has become somewhat of a de facto standard for robot KBs. However, as a general upper ontology encompassing a wide spectrum of human knowledge, OpenCyc frequently lacks domain-specific knowledge.

Ye et al. (2022) proposed ontology-enhanced prompt-tuning, pointing out that 'dataset-related ontologies' are expected for different tasks. For our task, we hypothesise that the KnowRob ontology, which supplemented OpenCyc with more thorough descriptions of concepts such as human everyday activities and household objects, may be a highly effective knowledge source eliminating the problem of knowledge noise. Much work has been done to build the KnowRob system over the past decade (Beetz et al., 2018). But being built on the Robot Operating System (ROS) and using SWI Prolog for knowledge inference raises some challenges in combining it with existing PLMs, which we address in this paper.

## 3. Methods

In this section, we present (a) our task formulation and relevant datasets (section §3.1), (b) the few-shot learning procedure (section §3.2) and (c) the process of knowledge-enhanced in-context learning (§3.3) in which three issues are addressed: (i) Ontology knowledge retrieval: retrieving action-object relevant ontology knowledge in KnowRob (§3.3.1); (ii) Texual knowledge representation: grounding knowledge to formal textual representations sent to GPT-3 (§3.3.2); (iii) Unseen knowledge learning with GPT-3: using GPT-3 to obtain knowledge assertions for terms for which the original KnowRob knowledge base does not provide explanations (§3.3.3)

### 3.1 Task and Datasets

Our action-effect prediction task takes English sentences describing the initial world $S_x$ and action description $S_a$ as input and generates another text $S_y$ describing the effects of its action on the objects it manipulates.

$$\underbrace{\{x_1, \cdots, x_k\}}_{S_x, precondition} + \underbrace{\{a_1, \cdots, a_m\}}_{S_a, action} \Rightarrow \underbrace{\{y_1, \cdots, y_n\}}_{S_y, post-condition} \tag{1}$$

| 1400 Action-Effect pairs | PIGLET |
|---|---|
| | **Precondition:** There is a cold apple on the counter. |
| **Action**: open door | **Action**: The robot slices the apple. |
| **Prediction:** The door is positioned so it can be passed through. | **Prediction**: The apple is now at room temperature, sliced and appears smaller. |

*Figure 2.* Examples of the action effect descriptions from two datasets: 1400 Action-Effect pairs and PIGLET. Colored words refer to terms with ontology knowledge descriptions: verbs, nouns, and objects in action descriptions are highlighted in red, green, and yellow, respectively. Terms in precondition descriptions are highlighted in blue.

Our task requires input and output of text. An action could be portrayed at the **word level** (verb-noun phrase) or **sentence level** (full sentence) depending on its complexity. Fig. 2 shows examples of these two levels of action descriptions.

**1400 Action-Effect pairs**: The 1400 action-effect pairs dataset created by Gao et al. (2018) can be directly used. In this dataset, for each action, its possible effects are described in natural language by 10 different annotators. So there are 140 actions in total. We randomly drew 8 actions as the training set, and the remaining 132 actions as the test set. The annotations for each action is in pair format : *verb_noun, effect_sentence*. Here, for each action phrase in the training set, we choose the longest sentence as the standard completion for in-context learning since the longer descriptions usually contain more information. We use the 10 annotations for each verb_noun phrase for evaluation.

**PIGLET**: PIGLET (Zellers et al., 2021) contains 2000 interactions (500 for training, 500 for validation and 1000 for testing) with English sentences describing the precondition, the action, and the action results.

### 3.2 Few-shot Learning with GPT

The procedure of applying a pre-trained GPT model to our text-to-text generation task is shown in the right half of Fig. 1. We aim to obtain action-effect predictions without fine tuning the parameters of LMs on our downstream task. Prompts are the only way to provide the task specification, so extensive *prompt engineering* is required to achieve high accuracy. We designed input prompts for GPT models.

**No-knowledge Prompt Design**: The second column of Table 1 shows templates prompts that do not entail external knowledge. We give the formatted prompts that we used for the two action-effect prediction datasets.

For the 1400 Action-Effect Pairs dataset, since the action description is just a verb-noun phrase. We use 'The person' for $[Connector1]$ and 'As a result,' for $[Connector2]$. These two connectors, along with the action phrase, provide the prompts for a single action. For the PIGLET dataset, both the precondition and action descriptions are in sentence form. To motivate the model to generate post-condition related descriptions, we employ 'Therefore,' as a $[Connector]$.

*Table 1.* Prompts Formats. $[S_k]$, $[S_x]$, $[S_a]$, and $[S_y]$ denote input slot for knowledge, precondition, action, and post-condition respectively. The second column shows the no-knowledge prompt template for our action prediction work. The third column shows corresponding knowledge-enhanced prompt formats.

| Dataset | No-knowledge Prompt Template | Knowledge-enhanced Prompt Template |
|---|---|---|
| 1400 Action-Effect Pairs | $[Connector1]$ $[S_a]$. $[Connector2]$ $[S_y]$ | $[S_k]$ $[Connector1]$ $[S_a]$. $[Connector2]$ $[S_y]$ |
| PIGLET | $[S_x]$ $[S_a]$ $[Connector]$ $[S_y]$ | $[S_k]$ $[S_x]$ $[S_a]$ $[Connector]$ $[S_y]$ |

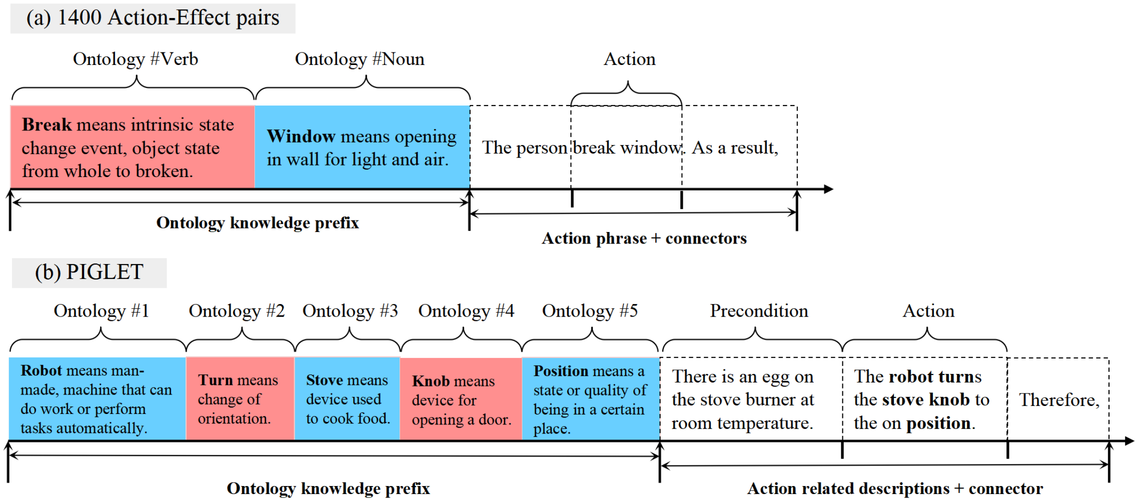

(a) 1400 Action-Effect pairs
Ontology #Verb
Ontology #Noun
Action
**Break** means intrinsic state change event, object state from whole to broken.
**Window** means opening in wall for light and air.
The person
break window
. As a result,
Ontology knowledge prefix
Action phrase + connectors
(b) PIGLET
Ontology #1
Ontology #2
Ontology #3
Ontology #4
Ontology #5
Precondition
Action
**Robot** means man-made, machine that can do work or perform tasks automatically.
**Turn** means change of orientation.
**Stove** means device used to cook food.
**Knob** means device for opening a door.
**Position** means a state or quality of being in a certain place.
There is an egg on the stove burner at room temperature.
The **robot turns** the **stove knob** to the on **position**.
Therefore,
Ontology knowledge prefix
Action related descriptions + connector

*Figure 3.* Examples of ontology knowledge-enhanced prompt for two action-effect prediction tasks.

**Knowledge-enhanced Prompts**: Since the generation of each word in a sequence is based on the word itself and all preceding words, we *prompt* the model with the knowledge statements by prepending $K$ before action related descriptions. Fig. 3 presents one complete instance for each dataset. The coloured boxes represent ontology knowledge obtained via Section 3.3, which are utilised as a prefix to the action-related descriptions. The dotted box depicts a no-knowledge prompt (only action relevant descriptions in prompt).

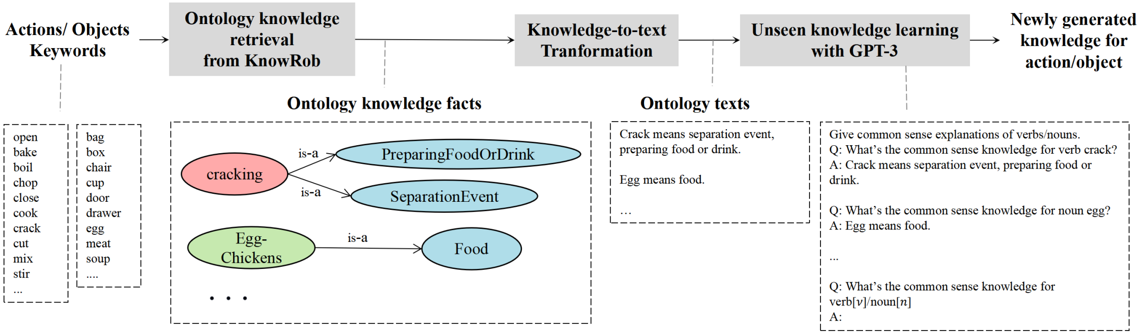
For each action $S_a = [verb, noun]$ (*e.g.*, 'cut potato') in the 1400 Action-Effect pairs dataset, we get two textual knowledge statements: $S_k = [k_{verb}, ..., k_{noun}]$. For PIGLET's sentence descriptions, we begin by extracting a concepts list $[c_1, c_2, ..., c_m]$ from action description sentence $S_a$ using matching 1-grams. For example, for sentence 'The robot turns the stove knob to the on position', the exact matching result would be [robot, turn, stove, knob, position]. Then, we prepend $m$ textual knowledge statements for $m$ tokens $S_k = [k_1, ..., k_m]$ before the precondition sentence $S_p$.

### 3.3 Knowledge Enhanced In-context Learning

We demonstrate our knowledge enhanced in-context learning method in this section. We show how we get ontology knowledge $[S_k] = [k_1, ..., k_m]$ and use these knowledge pieces to form the knowledge-enhanced prompts for GPT-3 to do in-context learning. We chose to build our knowledge base around KnowRob for our action-effect prediction task. Although the coverage of this knowledge base is limited, it provides clear, short explanations of actions and household objects that humans intuitively associate with the concepts.

#### *3.3.1 Ontology Knowledge Retrieval*

From 1400 action-effect pairs dataset, we got 62 verbs (action verbs) and 42 nouns (objects to be manipulated) in total. For PIGLET, we got an n-gram list $[c_1, c_2, ..., c_m]$ for each sentence. We retrieve ontology knowledge for the 62 verbs and 42 nouns from KnowRob.



Crack means separation event, preparing food or drink.

Egg means food.

...

Give common sense explanations of verbs/nouns.
Q: What's the common sense knowledge for verb crack?
A: Crack means separation event, preparing food or drink.

Q: What's the common sense knowledge for noun egg?
A: Egg means food.

...

Q: What's the common sense knowledge for verb[$v$]/noun[$n$]
A:

*Figure 4.* The pipeline of our knowledge processing method. There are three steps: (a) retrieve action-object relevant knowledge in KnowRob, (b) ground knowledge to formal textual representations, and (c) complete knowledge base $K$ with GPT-3 for unseen verbs or nouns.

In KnowRob (Tenorth & Beetz, 2013), Description Logic (DL) was used as a formalism to represent commonsense knowledge, in particular the OWL, which stores DL formulas in an XML-based file format. The OWL files can be queried using Prolog predicates. Among all OWL triples, the most common *Predicate* in the KnowRob ontology is *subClassOf.* As shown in the 'Ontology knowledge facts' box in Fig. 4, the concepts, *e.g.*, *PreparingFoodOrDrink*, *SeparationEvent*, *Food*,

*Table 2.* Symbolic ontology Knowledge to text examples. The ‘OWL’ column shows examples of OWL axioms in KnowRob.owl. The ‘Axiom’ column shows the rewritten axioms and class descriptions in the ACE controlled English language. The last column shows verbalization results using the ACE function words.

| OWL | Axiom | ACE text |
|---|---|---|
| <owl:Class rdf:about="&KnowRob;Mixing"> | | |
| <rdfs:subClassOf rdf:resource="&KnowRob;Incorporation-Physical"/> | | |
| <rdfs:subClassOf> | | |
| <owl:Restriction> | | |
| <owl:onProperty rdf:resource="&KnowRob;subAction"/> | | |
| <owl:someValuesFrom rdf:resource="&KnowRob;AddingSth"/> | SubClassOf(Class(#Mixing),Class(#Incorporation-Physical)) | |
| </owl:Restriction> | SubClassOf(Class(#Mixing), ObjectSomeValuesFrom( | Every Mixing is an Incorporation-Physical. |
| </rdfs:subClassOf> | ObjectProperty(#subAction),Class(#AddingSth))) | Every Mixing subAction an AddingSth. |
| <rdfs:subClassOf> | SubClassOf(Class(#Mixing), ObjectSomeValuesFrom( | Every Mixing subAction a Stirring. |
| <owl:Restriction> | ObjectProperty(#subAction),Class(#Stirring))) | |
| <owl:onProperty rdf:resource="&KnowRob;subAction"/> | | |
| <owl:someValuesFrom rdf:resource="&KnowRob;Stirring"/> | | |
| </owl:Restriction> | | |
| </rdfs:subClassOf> | | |
| </owl:Class> | | |

*Egg-Chickens*, are organised in a taxonomy, with *is-a* definitions describing relationships between them).

According to this observation, we query the KnowRob ontology base using the querying method *rdf_has(Subject, Relation, Object)* with verb and noun from action phrases as *Subject*. It returns *Object* matches for all specializations with Predicate defined as subPropertyOf. For 62 verbs and 42 nouns in this dataset, there are 10 verbs in the ontology [*open, bake, boil, chop, close, cook, crack, cut, mix, stir*] and 9 nouns [*bag, box, chair, cup, door, drawer, egg, meat, soup*] contained in KnowRob. Detailed retrieval ontologies with SWI-Prolog for the 10 verbs and 9 nouns can be found in Appendix A[2]. The retrieved ontologies provide the basis for ontology knowledge statements.

### 3.3.2 Texual Knowledge Representation

The retrieved knowledge obtained from the first step is structured ontology knowledge. The original OWL syntax (shown in ‘OWL’ column in Table 2) has class, property and individual names, which are connected to each other by various logical operators/functors (*e.g.*, rdfs:SubClassOf, owl:Class, owl:Restriction). We conduct two steps to transform the retrieved ontology extracts to natural language that works for GPT-3:

1. **Map all retrieved OWL ontology fragments to Attempto Controlled English (ACE) text** (Fuchs et al., 2008) – a subset of English that has a clear and unambiguous semantics in first-order logic. To do this, we use the OWL Syntax Converter and the OWL verbalizer created by Kaljurand (2007), which can provide a version of English that is comprehensible and acceptable. As a result of this step, all logic-specific words or symbols will be transformed into common words. We show the transformation for this step in Table 2. After mapping, the logical operators/functors are replaced with words (*e.g.*, ‘every’, ‘is a’).

2. **Transform ACE text to plain text.** After mapping KnowRob, the the ACE text contains many terms that are actually composed of several words (*e.g.*, PreparingFoodOrDrink), i.e. it capitalizes the first letter of each word in a string, following the CamelCase convention. To change such

[2] https://github.com/lfj95/action-effect-gpt/blob/main/appendix.pdf

forms into plain text, we insert a space before each uppercase letter, and convert all letters from uppercase to lowercase. Furthermore, the *subClass* descriptions ‘Every’ and ‘is a’ are removed; all texts containing the abbreviation ‘sth’ are replaced with ‘something’. When there is more than one axiom for a verb or noun, we concatenate all the texts to form a single “sentence” using a comma.

#### *3.3.3 Unseen Knowledge Learning with GPT-3*

After the previous step, we obtained ontology knowledge statements for 10 verbs and 9 nouns from KnowRob. The 1400 action-effect pairs dataset contains 62 verbs and 42 nouns in total. Every sentence in PIGLET has an n-gram list. How to obtain knowledge assertions for these terms? We address the challenge by utilising GPT-3 to learn previously unknown ontological knowledge.

For each verb or noun that has retrieved ontology knowledge, the sentences obtained from the above step are taken as training examples for GPT-3. As shown in Fig. 4, all retrieved knowledge for verbs and nouns form the “A” part of each Q-A pair. There are 19 example input-output pairs in total. Given the task description, verb/noun examples and their knowledge definitions, expressed as the translated sentences from the second step along with the query prompt, GPT-3 will generate its version of knowledge for a new word. We use this in-context learning method to generate knowledge descriptions for the remaining 52 verb and 33 nouns in 1400 action-effect pairs dataset and all word n-grams in the PIGLET dataset.

## 4. Experiments

In this section, we present: (i) unseen knowledge learning results (section §4.1); (ii) the few-shot learning experiments results (section §4.2). All the experiments were conducted on the two datasets introduced in Section 3.1; (iii) new state-of-the-art on PIGLET dataset with fine-tuning GPT-3 on KnowRob knowledge-enhanced inputs; (iv) example study.

### 4.1 Unseen Knowledge Learning with GPT-3.

In Table 3, we give several examples of knowledge learning results with GPT-3, which were randomly selected from a subset of novel generations concepts not found in the KnowRob ontology from the generation set for the two datasets. One of the authors conducted a manual evaluation to see whether ontology-based knowledge generation for verb or noun completes a plausible description. We can see that the generations are generally of high quality. The same author also checked all of the knowledge it generated for 52 verbs and 33 nouns in 1400 action-effect pairs dataset, and noted that almost all of the generations are informative except perhaps for “football means sport" “baseball means sport" and “scratch means a separating event". Detailed generation results can be found on our GitHub page. The derived knowledge is directly applied in the process of building knowledge-enhancement prompts.

*Table 3.* Examples of ontology-based knowledge learning results with GPT-3. Generations were randomly selected from a subset of novel generations from the 1400 action-effect pairs set and PIGLET. A novel generation is one which expresses knowledge not found in KnowRob.

| Actions | Knowledge learning results | Plausible |
|---|---|---|
| | **1400 Action-Effect pairs** | |
| bend knee | Bend means change in shape, usually over a distance. Knee means body part. | ✓ |
| bind hair | Bind means to connect or fasten together. Hair means body covering. | ✓ |
| crash car | Crash means sudden, uncontrolled motion. Car means a wheeled vehicle.. | ✓ |
| ignite wood | Ignite means create a flame. Wood means natural resource. | ✓ |
| lock cabinet | Lock means intrinsic state change event, object state from unlocked to locked. Cabinet means furniture, enclosed structure. | ✓ |
| | **PIGLET** | |
| The robot throws the statue across the room. | Statue means three-dimensional work of art. Room means space with walls and a ceiling, inside a building. Robot means man-made, machine that can do work or perform tasks automatically. Throw means motion event. | ✓ |
| The robot empties the bowl. | Bowl means eating utensil, concave container. Empty means without contents. Robot means man-made, machine that can do work or perform tasks automatically. | ✓ |
| The robot throws the keys a short distance. | Key means device used to open a lock. Short means describing object attribute. Distance means the property of being apart in space, the property of being far apart in time. Robot means man-made, machine that can do work or perform tasks automatically. Throw means motion event. | ✓ |

## 4.2 Prediction Results

We conducted few-shot learning ($K = 1, 2, 4, 8$). We use the designed prompts to demonstrate how the LM provides the answer to the actual prompt instantiated with the input. Example prompts can be found in Appendix B. We ran our experiments on four sizes of GPT-2 models (small, medium, large and XL with parameters 117M, 345M, 774M, and 1.5B) and GPT-3 models (Ada, Babbage, Curie and Davinci with parameters 2.7B, 6.7B, 13B, and 175B respectively). With a larger model size trained on more training data than GPT-2, GPT-3 is an enhanced version that can give improved prediction. Here, we want to explore how the size of the LM affects the results obtained and test whether adding knowledge to the prompt will enable GPT-2 to perform at a level comparable to that of GPT-3. We accessed GPT-2 through HuggingFace[3], and GPT-3 using the OpenAI API. For GPT-3, the sampling temperature $T$ determines how much randomness is in the output. To keep the results consistent, and to better judge the difference the knowledge-based prompt makes, we set $T = 0$, that is, argmax sampling. We generated greedily from the LM until it produced a full stop character. We evaluate the LM's performance with one automatic metric and one LM-based metric: Bleu(Papineni et al., 2002) and BERTScore (Zhang et al., 2019). Bleu measures the overlap between the generated responses and the ground truth. BERTScore evaluates text generation by computing token similarity with the BERT model.

Our approach for generating knowledge is compared to the following benchmarks on the two action-effect datasets:

1. **Baseline** - No external knowledge source. Inference with GPT-2 and GPT-3 with only precondition and action descriptions as the source of input.

[3] https://huggingface.co/gpt2, this and all other cited URLs last retrieved 2nd Nov.2022

*Table 5.* Few-shot learning results of GPT-2 and GPT-3 on on test sets of two datasets: 1400 Action-Effect pairs dataset and PIGLET dataset. We report BLEU and BERTScore F1 for $K$-shot learning $K = 1, 2, 4, 8$ (♯ examples from training set). The table's background colour represents various evaluation metrics: Green represents BERTScore, while blue represents BLEU score. Gradation of the colour indicates the level of the results; better results are shown in darker colour. In the rows with coloured numbers, red indicates a decrease and green an increase in the performance of our knowledge-enhanced method compared to the baseline.

| Dataset | Metrics | Knowledge | $K = 1$ | | $K = 2$ | | $K = 4$ | | $K = 8$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | GPT-2 | GPT-3 | GPT-2 | GPT-3 | GPT-2 | GPT-3 | GPT-2 | GPT-3 |
| 1400 Action-Effect pairs | BLEU | Baseline | 8.5 | 10.5 | 11.9 | 8.4 | 5.2 | 19.2 | 25.4 | **29.1** |
| | | Comet | 12.7 | 11.2 | 11.0 | 10.9 | 12.9 | 18.8 | 19.3 | 22.6 |
| | | KnowRob | **19.7** | **12.2** | **14.1** | **15.5** | **21.5** | **23.6** | **31.1** | 27.5 |
| | | | (+11.2) | (+1.7) | (+2.2) | (+7.1) | (+16.3) | (+4.4) | (+5.7) | (-1.6) |
| | BERT Score | Baseline | 61.2 | 61.6 | **65.3** | 60.0 | 66.9 | 67.0 | 76.1 | **74.3** |
| | | Comet | 65.0 | 62.0 | 62.2 | 60.7 | 67.5 | 67.0 | 73.6 | 69.7 |
| | | KnowRob | **70.0** | **63.9** | **65.3** | **66.2** | **73.8** | **71.4** | **78.5** | 72.8 |
| | | | (+8.8) | (+2.3) | (+0) | (+6.2) | (+6.9) | (+4.4) | (+2.4) | (-1.5) |
| PIGLET | BLEU | Baseline | 31.2 | 35.2 | **29.5** | 36.6 | 35.5 | 38.6 | 33.5 | 41.1 |
| | | ConceptNet | 33.3 | 33.6 | 25.2 | 35.7 | 36.0 | 41.3 | 38.6 | 41.1 |
| | | KnowRob | **34.7** | **37.3** | 29.4 | **37.7** | **37.0** | **41.9** | **39.5** | **42.6** |
| | | | (+3.5) | (+2.1) | (-0.1) | (+1.1) | (+1.5) | (+3.3) | (+6.0) | (+1.5) |
| | BERT Score | Baseline | 78.4 | 80.2 | 76.9 | 79.9 | 79.4 | 81.1 | 77.2 | 80.9 |
| | | ConceptNet | 78.8 | 78.6 | 74.4 | 78.9 | 79.1 | 81.5 | 79.5 | 81.0 |
| | | KnowRob | **78.9** | **80.7** | **77.0** | **80.5** | **79.5** | **82.1** | **80.0** | **81.4** |
| | | | (+0.5) | (+0.5) | (+0.1) | (+0.6) | (+0.1) | (+1.0) | (+0.8) | (+0.5) |

2. **ConceptNet** - Template-generated knowledge clarifications get from ConceptNet. First, we extract relation paths between words from the precondition description and words from the the action description. Then we convert each ConceptNet relation to a natural language template as in (Shwartz et al., 2020).
3. **Comet** - COMmonsEnse Transformers (COMET) (Bosselut, Rashkin, Sap, Malaviya, Celikyilmaz, & Choi, 2019). The model was trained on a seed set of knowledge tuples from ConceptNet (*subject*, *relation*, *object*) to generate commonsense descriptions. To use this KB to our action-effect prediction task, first, we take the action phrases as (Subject); for Relation, we consider *Causes*. We then generate (Object) for given action Subject and effect relevant Relations as inputs. Finally using beam search, ten candidates are produced for each action and relation type.

In Table 5, we show the few-shot learning results of GPT-2 and GPT-3 on test sets of two datasets: 1400 Action-Effect pairs dataset and PIGLET dataset. There is a trend of improving results when the in-context learning examples increases. In most cases, the prediction performance improves when exploiting completed ontology knowledge from KnowRob, especially for 4-shot learning. There is a gain of over 10 BLEU points for GPT-2 and 4.4 BERTScore for GPT-3 on 1400 action-effect pairs dataset and 3.3 BLEUScore for GPT-3 on PIGLET datset. The experimental results show that our KnowRob ontology knowledge enhanced method can achieve the best performance on both datasets: 8-shot learning with GPT-2 on 1400 Action-Effect pairs dataset and 8-shot learning with GPT-3 on PIGLET.

*Table 6.* Comparison of different knowledge-enhanced methods on GPT-2 and GPT-3 with different size in terms of BERTScore. Gradation of the colour red indicates the level of the results; better results are shown in darker red. The optimal performance settings for GPT-2 and GPT-3 are displayed in bold.

| Database | Knowledge | GPT-2 | | | | GPT-3 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Small | Medium | Large | XL | Ada | Babbage | Curie | Davinci |
| 1400 Action-Effect pairs | Baseline | 72.3 | 69.5 | 75.9 | 76.1 | 64.9 | 72.2 | 74.3 | 70.6 |
| | Comet | 73.0 | 69.6 | 72.0 | 73.6 | 62.1 | 68.7 | 69.7 | 67.2 |
| | KnowRob | 68.6 | 74.9 | 76.4 | **78.5** | 63.8 | 72.3 | **72.8** | 69.6 |
| PIGLET | Baseline | 76.7 | 77.7 | 78.5 | 77.2 | 76.7 | 78.9 | 80.2 | 80.9 |
| | ConceptNet | 77.3 | 78.7 | 79.6 | 79.5 | 75.6 | 78.3 | 79.7 | 81.0 |
| | KnowRob | 77.5 | 79.7 | **80.2** | 80.0 | 77.5 | 79.5 | **81.5** | 81.4 |

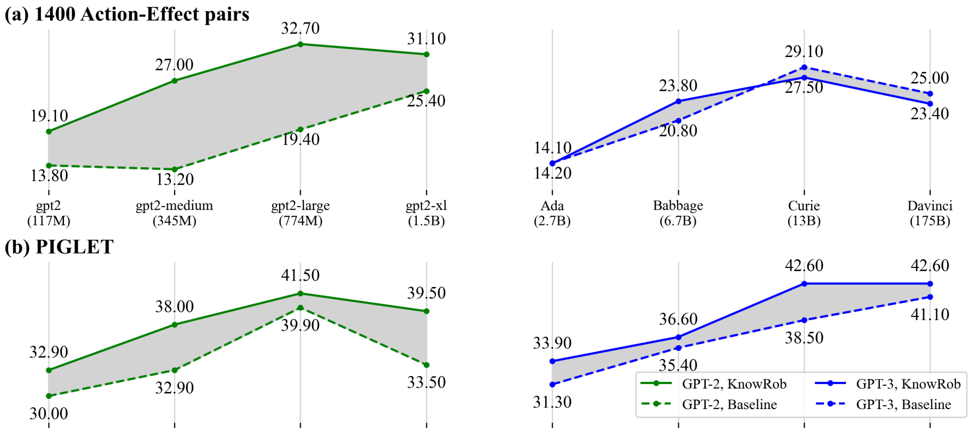


*Figure 5.* BLEU score with $K = 8$ KnowRob prompt prefixes and pure LM model on PIGLET for different sizes of GPT-2 and GPT-3.

**Model size matters.** Table 6 and Fig. 5 shows the performance gain when using different size of GPT-2 and GPT-3 as the post-condition prediction model. On the whole, there is an increasing trend in BLEU and BERT performance scores when model size for GPT-2/ GPT-3 grows (more apparent in BERTScore and PIGLET).

For the 1400 action-effect pairs dataset, the GPT-2 model with the highest BERTScore is GPT2-XL, whereas GPT2-Large has the highest BLEU score. Curie (the second largest GPT-3 model) got both highest BERTScore and BLEU score among all GPT-3 models. In both scores, GPT2-XL outperforms Curie for this dataset.

The best results for PIGLET are obtained on the second largest size for both GPT-2 and GPT-3: GPT2-XL for GPT-2 and Curie for GPT-3. Curie outperforms GPT-2XL with a 1.3 improvement in BERTScore. With Knowrob-enhanced few-shot learning, the differences between the largest and second-largest in BERTScores are negligible. But the BLEU score decreases by six points when the model size is changed from GPT2-Large to GPT2-XL.

### 4.3 Few-shot Learning & Fine-tuning

Apart from knowledge-enhanced few-shot learning, we also conducted fine-tuning experiments. We compare the KnowRob knowledge-enhanced few-shot ($K = 8$) learning results and fine-tuning results of GPT-3 (Davinci) model with the PIGPeN-NLG (Zellers et al., 2021) baseline in Table 7.

- **PIGPeN-NLG.** The model uses object states rather than using precondition textual descriptions. Given the object lists, each object involved is represented with 42 attributes. The model is based on GPT-small (117M parameters).
- **KnowRob-GPT3-8shot.** We do in-context learning on the Davinci model with 8 randomly selected training examples. We use the KnowRob knowledge as prefix to precondition and action texts.
- **KnowRob-GPT3-finetune.** We fine-tune the Davinci model on 500 training examples of PIGLET. The KnowRob knowledge prefix is concatenated with precondition and action texts as prompt. the post-condition texts are used as completion label.
- **Human.** The results comes from Zellers et al. (2021). From their code released on GitHub [4], there are three sets of human annotations for each sample, They used one set as human predictions, and the other two formed the "reference" set (baseline post-condition descriptions) used for evaluation of both the human and machine predictions.

*Table 7.* Post-condition text generation results comparison on validation and test set of PIGPeN-NLG. Finetuning GPT-3 (Davinci) with KnowRob enhanced action descriptions outperforms the state-of-the-art PIGLET model by 7 BLEU points and 3 BERTScore $F_1$ points.

| Model | BLEU | | BERTScore | |
|---|---|---|---|---|
| | Val | Test | Val | Test |
| PIGPeN-NLG(Zellers et al., 2021) | 49.0 | 43.9 | 83.6 | 81.3 |
| KnowRob-GPT3-8shot | 42.2 | 42.6 | 81.3 | 81.4 |
| KnowRob-GPT3-finetune | **56.3** | **58.6** | **86.2** | **86.7** |
| Human | 44.5 | 45.6 | 82.6 | 83.3 |

The experimental results show that the ontology knowledge enhanced in-context learning, with just only 8 examples from training set as input, can achieve similar level (only 1.3 BLEU points lower and 0.1 BERTScore F1 points lower) of PIGPeN-NLG performance, which are trained on 500 training examples. Furthermore, finetuning the GPT-3 model on all 500 training examples with ontology knowledge produce a new state-of-the-art result: an increase of 4.7 BLEU points and 4.6 BERTScore F1 points on the PIGLET test set, even higher than human predictions. The BLEU and BERTScore shown in Table 7 are calculated using the same reference set. These two automatic metrics are efficient methods of benchmarking, but clearly have potential deficiencies. In particular, some generated statements which are in fact appropriate descriptions of the action effect,

[4] https://github.com/rowanz/piglet

may receive a low score, because the scores are based only on a set of reference sentences (in our experiments. one set of human annotations). Although a high score suggests that the generation predictions coincide closely with the set of human annotations used for evaluation, the converse does not necessarily mean that the prediction would not be in agreement with some other human annotator.

## 4.4 Qualitative Analysis

In Table 8, we report five randomly selected 4-shot learning results from the Davinci model for each dataset. Both with and without knowledge predictions are reasonable under Davinci's 4-shot learning, and KnowRob achieved the highest improvement over baseline on both datasets in this setting, which can help us analyse what impact the imported KnowRob knowledge has on post-condition prediction.

*Table 8.* Example 4-shot learning results with Davinci using the question answering in context template. The 'Label' column denotes manually collected action effect descriptions. The 'No-knowledge generations' column shows the generation results with only action input. The 'KnowRob-enhanced generations' column shows the generation results using action phrases together with knowledge learnt from KnowRob as input.

| **Action** | **Label** | **No-knowledge generations** | **KnowRob-enhanced generations** |
|---|---|---|---|
| **1400 Action-Effect pairs** | | | |
| bake potato | the potato is heated in the oven so it can be eaten | the potato is heated up to a high temperature and then undergoes a change in state from solid to gas | the potato is heated to a temperature that causes it to transform from its initial raw state to a cooked state |
| bend knee | fold the leg in a way the knee is bent forward | the knee is bent in a way that increases the angle between the thigh and shin bones | the knee is bent to a certain angle |
| cook onion | an onion is heated in an oven or on a stove to make it ready to eat | the onion is heated and its enzymes are denatured | the onion is heated and the skin is removed |
| kick door | the door lock is broken and swings open | the door is hit with the foot and moves backwards | the door is propelled away from the person and opens |
| bite apple | a mouth uses teeth to remove a section of apple from itself | the apple is bitten into and the edible flesh is consumed | the apple is bitten into with force and the bitten off piece is removed from the apple |
| **PIGLET** | | | |
| The robot is holding a glass bottle in front of it. The robot throws the bottle across the room. | The bottle breaks into little shards of glass, it is no longer picked up any more, and it is now farther away. | The bottle is now broken. | The bottle is now broken and the room is now messy. |
| There's a plate in an open cabinet and the robot is holding a cup. The robot closes the cabinet. | The cabinet is now closed. | The plate is now inside the cabinet. | The cabinet is now closed and the plate is inside. |
| The robot is standing in front of a stove with a pan on it that is turned off. The robot turns on the stove. | The stove gets hot and the pan gets hot. | The pan is now on the stove. | The stove is now turned on and the pan begins to heat up. |
| The robot is standing in front of a drawer that is closed and there is bread on the counter. The robot opens the drawer. | The drawer is now open. | The bread is now in the drawer. | The drawer is now open and the bread is inside the drawer. |
| The robot is in front of a full sink with a running faucet. The robot turns off the faucet. | The water stops and the sink is now empty of liquid. | The faucet is now off and the liquid in the sink is no longer flowing. | The faucet is turned off and the sink is empty. |

For the 1400 action-effect pairs dataset prediction examples, the human annotations are concise in all cases. While for some cases, it put more focus on the action process rather than post-condition description. (*e.g.*, for 'bend knee', 'fold the leg in a way'; for 'bite apple', it produced descriptions such as 'a mouth uses teeth to remove') We can see that with Davinci, both generations are acceptable . The generations without a knowledge prompt tend to contain some not everyday technical terms (*e.g.*, for 'cook onion', it produced text such as 'enzymes are denatured'; for 'bite apple', it mentioned 'edible flesh'.) Additionally, actual factual errors are made (*e.g.*, for 'bake potato', it says potato 'undergoes a change in state from solid to gas'). The KnowRob knowledge-enhanced learning results of these examples are easier to understand (e.g., for 'bite apple', it stated 'the bitten off piece is removed from the apple' rather than 'edible flesh is consumed').

For PIGLET, the human annotations are often brief, but include the most significant post-condition information. Generations without knowledge prompts usually explain the state changes of a single object, but tend to be impacted by things that occur in precondition descriptions, but have nothing to do with action. For instance, in the second example 'closes the cabinet', apart from cabinet, the precondition descriptions also mentioned 'holding a cup' and 'a plate in cabinet', the prediction 'plate inside the cabinet' actually is not the effect of 'closes the cabinet' but put the plate in cabinet. The post-conditions generated by GPT-3 with KnowRob-enhanced in-context learning are sometimes also affected by unimportant things but they include the essential post-condition descriptions. (*e.g.*, 'cabinet is now closed' apart from 'plate is inside' for 'close cabinet', 'drawer is open' apart from 'bread inside drawer' for 'open the drawer').

## 5. Conclusions

We proposed an action-effect prediction task, which unlike existing causality reasoning tasks, focuses on generation rather than doing multiple choice selection. Such a generation task has a broader application, e.g., robot manipulation, and combining this textual reasoning with visual information for further reasoning (Li et al., 2022). We employed pre-trained GPT models to solve this task. At the same time, we improved such LM's reasoning ability by injecting external knowledge from a strongly relevant ontology KB: KnowRob. We form a bridge from a symbolic knowledge base to a LM by converting ontologies to textual prompts, thus demonstrating a new use for traditional KBs.

In this work, while we use KnowRob as the main KB, we also make a comparison with ConceptNet and Comet. This can be extended to other KBs, e.g., VerbNet, Wiki-data, web-searching knowledge. This points to future work in extending the approach to a variety of other types of KBs, as well as more efficient methods of embedding the knowledge into pre-trained LMs for commonsense reasoning. Furthermore, if in future work we could translate the language predictions back to a formal ontology, this would allow automated ontology enhancement by exploiting the knowledge implicit in large scale LMs. In addition, while PLMs have the advantage of being able to produce predictions from varied input, they also have a disadvantage: the outcomes are difficult to explain. Thus developing explainable LMs would seem to be a critical research topic for allowing LMs to be deployed in cognitive systems. Also, for our experiments, to save token usage and time, we randomly chose 8 examples from 1400 action-effect dataset as training examples, and randomly chose $K$ examples from the training sets training examples for $K$-shot learning; it would be interesting to try different sets of training examples and do cross-validation experiments to see whether the example selection has any impact on the results. Finally, in this work we mainly focus on sentence-level action-effect prediction, we hope that this will pave the way for further research into more complicated tasks (e.g., quantity change prediction, status tracking, action planning). and other commonsense reasoning tasks.

## References

Alomari, M., Li, F., Hogg, D. C., & Cohn, A. G. (2022). Online perceptual learning and natural language acquisition for autonomous robots. *Artificial Intelligence*, *303*, 103637.

Beetz, M., Beßler, D., Haidu, A., Pomarlan, M., Bozcuoğlu, A. K., & Bartels, G. (2018). Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents. *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 512–519). IEEE.

Bian, N., Han, X., Chen, B., & Sun, L. (2021). Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 12574–12582).

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). Comet: Commonsense transformers for automatic knowledge graph construction. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4762–4779).

Brown, T., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Cox, M., Dannenhauer, D., & Kondrakunta, S. (2017). Goal operations for cognitive systems. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dalvi, B., Tandon, N., Bosselut, A., Yih, W.-t., & Clark, P. (2019). Everything happens for a reason: Discovering the purpose of actions in procedural text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 4496–4505).

Davis, E. (1998). The naive physics perplex. *AI magazine*, *19*, 51–51.

Fine-Morris, M., Auslander, E. B., Floyd, M. W., Pennisi, G., Muñoz-Avila, H., & Gupta, E. K. M. (2020). Learning hierarchical task networks with landmarks and numeric fluents by combining symbolic and numeric regression. *Proceedings of the 8th Annual Conference on Advances in Cognitive Systems*.

Fuchs, N. E., Kaljurand, K., & Kuhn, T. (2008). Attempto controlled English for knowledge representation. In *Reasoning web*, 104–124. Springer.

Gao, Q., Yang, S., Chai, J., & Vanderwende, L. (2018). What action causes this? Towards naive physical action-effect prediction. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 934–945).

Kaljurand, K. (2007). *Attempto controlled English as a semantic web language*. Doctoral dissertation, Faculty of Mathematics and Computer Science, University of Tartu.

Kapanipathi, P., et al. (2020). Infusing knowledge into the textual entailment task using graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 8074–8081).

Kirk, J. R., Wray, R. E., Lindes, P., & Laird, J. E. (2022). Evaluating diverse knowledge sources for online one-shot learning of novel tasks. *arXiv preprint arXiv:2208.09554*.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*, 33–38.

Li, F., Hogg, D. C., & Cohn, A. G. (2022). Exploring the glide model for human action-effect prediction. *P-VLAM*, (p. 1).

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, Y., Wan, Y., He, L., Peng, H., & Philip, S. Y. (2021b). KG-BART: Knowledge graph-augmented BART for generative commonsense reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 6418–6425).

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 8086–8098).

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

Olmo, A., Sreedharan, S., & Kambhampati, S. (2021). GPT3-to-plan: Extracting plans from text using GPT-3. *FinPlan 2021*, (p. 24).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318).

Rashkin, H., Sap, M., Allaway, E., Smith, N. A., & Choi, Y. (2018). Event2mind: Commonsense inference on events, intents, and reactions. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 463–473).

Shwartz, V., West, P., Le Bras, R., Bhagavatula, C., & Choi, Y. (2020). Unsupervised commonsense question answering with self-talk. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4615–4629).

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. *Thirty-first AAAI conference on Artificial Intelligence*. AAAI Press.

Tandon, N., Dalvi, B., Grus, J., Yih, W.-t., Bosselut, A., & Clark, P. (2018). Reasoning about actions and state changes by injecting commonsense knowledge. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 57–66).

Tenorth, M., & Beetz, M. (2013). Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, *32*, 566–590.

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*, 78–85.

Wolf, T., et al. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38–45).

Wray, R. E., Kirk, J. R., Laird, J. E., et al. (2021). Language models as a knowledge source for cognitive agents. *arXiv preprint arXiv:2109.08270*.

Wu, Y., Zhao, Y., Hu, B., Minervini, P., Stenetorp, P., & Riedel, S. (2022). An efficient memory-augmented transformer for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2210.16773*.

Xiao, J., Shang, X., Yao, A., & Chua, T.-S. (2021). NEXT-QA: Next phase of question-answering to explaining temporal actions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9777–9786).

Ye, H., Zhang, N., Deng, S., Chen, X., Chen, H., Xiong, F., Chen, X., & Chen, H. (2022). Ontology-enhanced prompt-tuning for few-shot learning. *Proceedings of the ACM Web Conference 2022* (pp. 778–787).

Zellers, R., Holtzman, A., Peters, M. E., Mottaghi, R., Kembhavi, A., Farhadi, A., & Choi, Y. (2021). Piglet: Language grounding through neuro-symbolic interaction in a 3D world. *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)* (pp. 2040–2050).

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). BERTScore: Evaluating text generation with BERT. *International Conference on Learning Representations*.

Zhou, X., Zhang, Y., Cui, L., & Huang, D. (2020). Evaluating commonsense in pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 9733–9740).