symmetry

MDPI

Article

Utilizing Language Models to Expand Vision-Based Commonsense Knowledge Graphs

Navid Rezaei [1] and Marek Z. Reformat [1,2,*]

[1] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
[2] Information Technology Institute, University of Social Sciences, 90-113 Łódź, Poland
* Correspondence: reformat@ualberta.ca

Abstract: The introduction and ever-growing size of the transformer deep-learning architecture have had a tremendous impact not only in the field of natural language processing but also in other fields. The transformer-based language models have contributed to a renewed interest in commonsense knowledge due to the abilities of deep learning models. Recent literature has focused on analyzing commonsense embedded within the pre-trained parameters of these models and embedding missing commonsense using knowledge graphs and fine-tuning. We base our current work on the empirically proven language understanding of very large transformer-based language models to expand a limited commonsense knowledge graph, initially generated only on visual data. The few-shot-prompted pre-trained language models can learn the context of an initial knowledge graph with less bias than language models fine-tuned on a large initial corpus. It is also shown that these models can offer new concepts that are added to the vision-based knowledge graph. This two-step approach of vision mining and language model prompts results in the auto-generation of a commonsense knowledge graph well equipped with physical commonsense, which is human commonsense gained by interacting with the physical world. To prompt the language models, we adapted the chain-of-thought method of prompting. To the best of our knowledge, it is a novel contribution to the domain of the generation of commonsense knowledge, which can result in a five-fold cost reduction compared to the state-of-the-art. Another contribution is assigning fuzzy linguistic terms to the generated triples. The process is end to end in the context of knowledge graphs. It means the triples are verbalized to natural language, and after being processed, the results are converted back to triples and added to the commonsense knowledge graph.

Keywords: commonsense; knowledge graph; linguistic terms; language models; deep learning

check for updates

Citation: Rezaei, N.; Reformat, M.Z. Utilizing Language Models to Expand Vision-Based Commonsense Knowledge Graphs. *Symmetry* **2022**, *14*, 1715. https://doi.org/10.3390/sym14081715

Academic Editors: Kóczy T. László and István A. Harmati

Received: 22 June 2022
Accepted: 11 August 2022
Published: 17 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

There has been a renewed interest in commonsense as a stepping stone toward achieving human-level intelligence. Some of the new research has shown how important commonsense knowledge graphs can be in training artificial intelligence (AI) models, which exhibit commonsense [1,2].

Commonsensical concepts should be symmetric to any changes in their representation. In the case of an ideal commonsense knowledge graph and an ideal language model, transforming concepts between the two representations of knowledge should not change their meaning. By an ideal language model, we mean a language model that is sufficiently large and capable that can understand language and all the concepts within. At the same time, an ideal commonsense knowledge graph is a knowledge graph that contains all correct commonsensical concepts.

The knowledge-symmetric transformation depends on the architecture of the language model and the knowledge graph, both of which are not ideal. These issues make deriving a transformation process that symmetrically maps knowledge from one to the other challenging and impractical. To compensate for this, we introduce a prompting methodology based on questions and answers to extract from the language model the knowledge missing in the knowledge graph. In that way, the symmetry of concepts is preserved by mapping them between two knowledge storage paradigms.

The main building blocks of knowledge graphs used to represent commonsense knowledge are subjects, predicates, and objects. Subjects and objects are other words for the nodes of the graph. The tail of a relationship is called an object, and the head is called a subject. The directed edge connecting the two is called a predicate. Knowledge graphs are directed heterogeneous graphs in some sense.

Artificial intelligence (AI) models are reported to have limited commonsense abilities [3,4]. Acquiring commonsense by AI systems can make the sample efficient in adapting to new environments, as proposed in [5]. Commonsense knowledge graphs can help AI systems both explicitly and implicitly: explicitly by querying the commonsense knowledge graph itself, or implicitly by knowledge transfer methods, such as the fine-tuning of language models as reported in [1]. This is similar to how a BERT model is fine-tuned on a SQuAD dataset for reading comprehension [6]. In addition, expressing commonsense knowledge in the symbolic format can help with commonsense knowledge explainability and the vetting process.

Using only vision to generate commonsense knowledge as proposed in [7,8] has its advantages and disadvantages. By processing images and videos, we can perceive visual cues that are not usually written or spoken about, but they make our common understanding of how physical entities exist and interact. On the other hand, fine-tuned vision-based deep learning models are limited to the concept and relation vocabulary that they are trained on, which is usually limited, and are not usually capable of understanding the intricacies of natural language. An ideal self-supervised vision model, which can absorb and learn all the visual interactions, could theoretically suffice. However, the current vision models have shortcomings that we believe can be addressed via the utilization of language models. For example, scene graph generation models are limited regarding the number of detected relationship types. Increasing the number of relationship types does not provide a satisfying solution, as there is still a bias toward the frequently seen relationships in the supervised training [9].

In this paper, we explore and use the extra knowledge that language models offer to expand on the limited auto-generated vision-based commonsense knowledge graphs. We chose to use few-shot learning in larger transformer-architecture-based language models, as larger models have shown to perform well on language benchmarks without requiring further fine-tuning on a specific task. We experiment with not only adding new concepts to the vision-based commonsense knowledge graph but also new types of relationships with fuzzy-style linguistic weights.

1.1. Commonsense Definition

Having a good definition of commonsense is imperative to better understand and discuss the work and results. Commonsense is simple, as almost everyone knows it, and is challenging, as no one often talks or writes about it.

Yann LeCun, an inventor of convolutional neural networks, believes that a collection of models of the world that represents what is likely, plausible, or impossible makes our commonsense [3]. John McCarthy classifies human commonsense into two categories of knowledge and ability. The commonsense ability is the action based on the gained commonsense knowledge [10].

Commonsense knowledge is inherently uncertain and context-dependent. The degree of correctness of commonsense knowledge depends on the common group of observers. For instance, the people who live in the northern hemisphere know July to be a hot summer month, while the people in the southern hemisphere observe it as a colder winter month.

Commonsense can also be classified into different topics, such as physical interactions, order of events, and social dynamics. In this paper, we mainly focus on physical commonsense, such as the usage of an object and its relative location, compared to other objects.

In a nutshell, commonsense knowledge graphs are graphs that represent facts and relations between them that are characteristic of real-world scenarios and situations. Such graphs focus on elements and aspects related to everyday activities, arrangements of things, and normal/natural circumstances, such as *flower in vase*, *tree has trunk*, *food on plate*, *shoe is less likely made of metal*, or *arm is most likely to be able to move, bend and be strong*.

Such facts seem very obvious and normal/natural for a human being, but this knowledge is not easy to be acquired by a machine. The gap in the processes of learning that type of information is filled out by techniques and methods linked to collecting and representing commonsense knowledge.

1.2. Contributions

The goal is to utilize transformer-based language models to expand vision-based commonsense knowledge graphs. In this paper, we propose an extension of the methodology for constructing a commonsense knowledge graph proposed in [7,8] with a technique based on questions and answers prompting very large language models. The new technique addresses generating prompts that are used as inputs to the language models. First, a prompt is entered into the model. Then, the obtained response that contains facts/information is added to expand a knowledge graph. The method is illustrated in Figure 1.

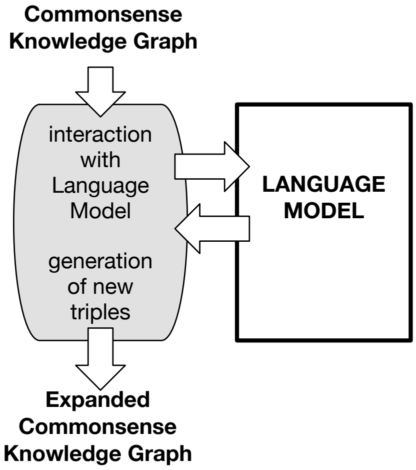
Commonsense Knowledge Graph
interaction with Language Model
generation of new triples
LANGUAGE MODEL
Expanded Commonsense Knowledge Graph

Figure 1. Expansion of a vision-based commonsense knowledge graph with relevant but new information.

In particular, the contributions of the paper are as follows:

- A multi-modal methodology for constructing commonsense knowledge graphs;
- A process of generating question/answer-based prompts for language models based on triples extracted from an existing commonsense knowledge graph, or based on the input from users;
- An expansion of the standard structure of knowledge graphs by introducing an approach to add degrees of likeliness as indicators of the 'strength' of triples that are added to commonsense knowledge graphs; the degrees are expressed with linguistic terms , such as **more likely**, **less likely**;
- An evaluation process based on Amazon Mechanical Turk.

2. Related Work

The work presented in this paper falls into the category of tasks that focus on completion and expansion of a commonsense knowledge base. There is related literature that addresses the methods and tools to achieve these goals.

2.1. Expansion of Knowledge Bases

The work presented in [11,12] focuses on link prediction between known entities within a graph. These methods are not able to expand beyond the current conceptual knowledge in the graph. They are more suited toward finding possible relations between currently known concepts.

Recent works have tried to use language models, especially transformers [13], to achieve better results in the tasks of the completion and expansion of the knowledge base. The authors of [14] use language models to construct knowledge graphs: they assume to have a subject and object and then use the language model to predict an appropriate relationship between them.

Ref. [15] indicates that using the next token prediction capability of pre-trained language models, one can use them as a factual knowledge base, e.g., to find the birthplace of a specific person. Among the language models analyzed, the largest transformer-based language model, BERT-Large [16], performed better than others. This paper confirms the overall consensus in the research community that the larger the language models become, the more capable they become.

There has been recent works to train very large generic transformer-based language models, such as GPT-3 by OpenAI [17], Meta's OPT-175B [18], and Google's PaLM [19]. There is a common consensus among all recent findings that larger language models can potentially be more capable of performing diverse tasks. Additionally, they do not need costly fine-tuning and data collection. Yet, providing appropriate prompts to language models can be challenging.

Processes of generating prompts are a subject of recent research publications. Prompts serve as input to large language models [20] and are used to reduce the amount of data required for fine-tuning [21]. By prompt, we mean a set of tokens and a short text that constitute the input to the model. Prompts could have different purposes, such as providing context, tone, or a sample of expected responses. They are part of the few-shot-learning process and are usually used instead of fine-tuning a language model. While prompts benefit the overall performance, their design does not follow a specific rule. Some even call the process 'prompt engineering.' Question and answering tasks are improved by few-example prompts when using large language models [17]. Extra chain-of-thought language prompts that contain reasoning steps are shown to improve more complex tasks related to arithmetics and commonsense [20]. The chain-of-thought process helps find missing parts of knowledge [22]. The work is similar to unsupervised data creation [23]. However, questions and answers used in this paper serve as prompts to foundation models. They are not used directly on text for reading comprehension.

2.2. Construction and Expansion of Commonsense Knowledge

A body of literature [1,2] focuses on annotating commonsense knowledge graphs to train language models for predicting commonsense information based on the given subject and predicate. The human-annotated knowledge graphs are typically in the size of millions and cover social interactions, events, and entity commonsense. The ATOMIC-COMET work [1] is based on manually creating a commonsense knowledge graph. This graph is used to train a small language model, such as GPT-2, on the human-annotated data. Our approach is different. We focus on generating a commonsense knowledge graph automatically rather than manually. The method comprises two phases, the first based on vision and the second enriching the results using language. The manual generation of commonsense knowledge graphs can become costly, as shown in [21]. Our approach seems to be more similar to [2], where GPT-3 is utilized to generate commonsense knowledge graphs. The proposed method is different in multiple ways. One is that we use a two-step method, where the feed for GPT-3 is provided by visual data, while [2] uses human-annotated data. Moreover, [2] only generates the most probable results, while we generate both highly probable and less probable results. Our approach has a cost of roughly one-fifth of the method described in [2] when considering the linguistic generation part and using the same GPT-3 model size. The reduced cost is because our prompt method accommodates the generation of $N = 5$ triples in one pass. Another difference is that [2] is proposed to only find an object given the subject and predicate, while our approach works in both ways and can suggest an appropriate subject given the predicate and object.

Several papers have experimented with the new very large transformers, such as GPT-3 [2]. This work focuses on prompting GPT-3 with some annotated commonsense triples, then extracting GPT-3's commonsense and adding it to a graph. It assumes predefined predicates and does not explicitly discuss the weight of the triples. The process takes subjects and predicates and uses causal language models to predict the most suitable objects. The method we introduce in this paper can also predict the triples' subjects.

It was discussed in [24,25] that training a model on commonsense knowledge base completion (CKBC) task suffers from low-coverage training data. Therefore, training on specific data results in the model's over-fitting and reduces its performance on novel data. Based on these observations, we focused our efforts on generically trained language models, which are large enough to accommodate few-shot learning.

A few recent works report on generating knowledge graphs from visual data. As an example, the NEIL method [26] extracts object relationships in images and results in 10,000 triples using 10 types of predicates.

One of the main physical commonsense knowledge graphs is ConceptNet [27]. As much as it can be helpful and treated as a reference, it has some drawbacks that our work can potentially resolve in the future. First, ConceptNet is mainly human annotated and cannot be continuously and cost-effectively updated. Our work suggests a methodology to continuously and automatically update the missing commonsense knowledge. Second, ConceptNet has a limited predicate related to location–the vague *AtLocation* predicate. Our method is able to enrich the commonsense knowledge with more fine-grained relations, such as *Above*, *Below*, and others. Third, ConceptNet is limited in terms of its predicate types, too. Our approach can enrich ConceptNet with new types of predicates, such as *NotIsA* or *CanEat*. An essential weakness of ConceptNet is its lack of context. For example, finding a desk in a classroom is more probable than in a bar. Our approach can potentially expand and enrich ConceptNet with weighted contextual relations. Moreover, it can be done automatically if part of ConceptNet is used as a seed commonsense knowledge graph.

3. Image-Based Construction of Commonsense Knowledge Graph

In our previous works, we introduced methodologies to generate a commonsense knowledge graph, called *world-perceiving knowledge graph* (*WpKG*), by only using visual data [7,8]. Like human infants who gain commonsense details about their physical world before they learn to express them in language, the introduced process focuses on deducing commonsense knowledge by observing many images.

The *WpKG* paper [7] introduces a methodology to auto-generate commonsense using deep learning models to perform object detection and relation prediction. The final *WpKG* graph has 7000 triples using 50 predicate and 150 entity types. [8] expands on the previous work to generate contextual and weighted commonsense knowledge graph, *C-WpKG*, in 93 contexts using state-of-the-art object and relation detection models. In the following sub-sections, we describe the process of reaching these results.

3.1. Extraction of Scene Graphs

The first step in the process is to analyze each image individually by detecting the existing objects and extracting possible relations between the objects in the image. The resulting graph representing objects in images as nodes and their relationships as edges is called a scene graph.

A convolutional neural network (CNN) model, such as Faster-RCNN [28], is used to detect the objects. To produce image features, ResNeXt-101-FPN CNN model [29] is utilized, which is needed for the region proposal network (RPN) of the Faster-RCNN model. The output of the pre-trained object detection model includes objects in the image, together with their bounding boxes and class scores.

To predict relations between the objects and generate a scene graph for each image, the MOTIFS model [30] unbiased by the Causal-TDE method [9] is used. Then, the scene graph for each image is generated based on the object features and relations between them.

3.2. Fusion of Scene Graphs

Regularly observed phenomena make up collective commonsense knowledge. Similarly, we aggregate the scene graphs extracted from the images into a single knowledge graph that comprises possible commonsense relations. To differentiate between relationships to know if a phenomenon is a one-time event or a typical one, we assign weights to the links representing the relations. Different methods of assigning weights to the observations are investigated. Among them, a probability-based approach is selected. It correlates the most with human commonsense during human evaluations. This weight assignment method follows Equation (1).

$$w_{t_i} = \sum_{j=1}^{|D_T|} \delta\left(t_i, t_j\right) \cdot P\left(t_j\right) \tag{1}$$

where w_{t_i} is a weight of the t_i triple, $\delta(\cdot)$ is Kronecker delta function, $P(t_j)$ represents the probability of detecting each instance of triple t_j, which is made of a subject (s), predicate (p), and object (o). The weights are also normalized by $\max\{w_{t_i} : t_i \in D_T\}$. The list of all detected triples is represented by D_T.

Variations of the same method have been shown to work in context-free and contextual scenarios. In this paper, we only focus on context-free visual commonsense knowledge.

4. Expanding Knowledge Graph Using Language Model

The automatic construction of commonsense knowledge graphs requires retrieving commonsense knowledge. It seems natural—also for a human being—to start that process by analyzing images and pictures representing real-world situations. Yet, to further increase commonsense knowledge and expand knowledge graphs, other sources of information are required and beneficial. One of them is verbal, textual information.

Therefore, to diversify information embedded in vision-based commonsense knowledge graphs and further expand them, we propose a human-like method of assimilating commonsense knowledge using linguistic-based data sources.

4.1. Methodology

The proposed method is intuitive and straightforward. It starts with interaction with a language model using short texts created based on the commonsense knowledge graph to be expanded. Then, the obtained results, i.e., the retrieved pieces of information and facts, are added to the graph as triples. The overview of the process is illustrated in Figure 2. It shows the *WpKG* as a graph from which some triples are extracted. The information from these triples is used to instantiate prompt templates (Section 4.3) that represent training data for a language model. The instantiated prompts are entered into the model. As a result, the obtained pieces of information are converted into new triples. These new triples are added to the *WpKG*, leading to its expansion.

4.2. Language Models

Larger language models, such as GPT-3, have shown promising results on diverse benchmarks with only a few examples of each task. The results are sometimes even comparable with smaller language models, which are fine-tuned on a large corpus of data. Recent research has shown the usefulness and effectiveness of large language models in automatic commonsense knowledge generation [2]. In this paper, we utilize different versions of a large language model, called GPT-3 [17], to expand vision-based commonsense knowledge graphs.

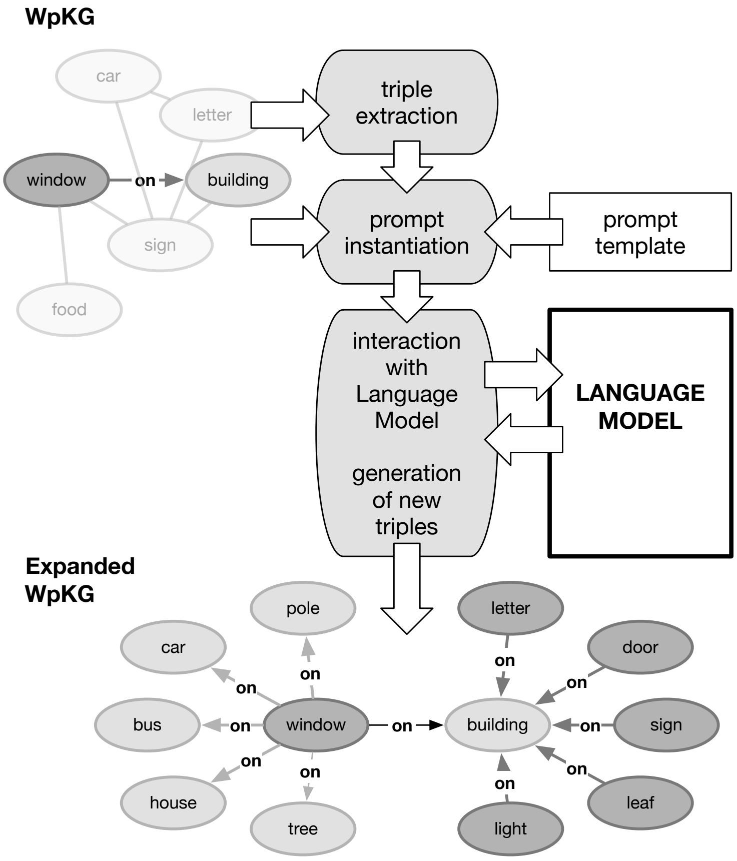

Figure 2. Process of expanding a graph using language model.

GPT-3 is a causal language model with almost the same yet larger architecture as previous iterations of the same model (GPT and GPT-2). The goal of a causal model is to predict the next token given the previous tokens. The language model assigns a probability to all the tokens to decide which one could happen next.

Choosing the highest probability next token may not be the best option, given the task. In this paper, we use nucleus (top-p) sampling to generate the text responses [31] and also adjust the temperature of the sampling to reach better results.

By reducing the temperature, we basically increase the likelihood of high-probability next tokens and reduce the likelihood of low-probability next tokens. This setting results in more deterministic next tokens to be chosen when selecting the next token randomly. The temperature is implemented as a coefficient inside the softmax function. Empirically, we observed that lower temperature works better for simpler cases, while the higher temperature can work for more complex cases that need diverse results, e.g., finding objects that are less likely to exist given a subject and a predicate.

In nucleus (top-p) sampling, instead of sampling from all the tokens, the algorithm chooses from the set of tokens that their cumulative probability of occurrence next is smaller than a given probability p. In our experiments, we keep the p value equal to one to choose from the most diverse vocabulary possible.

4.3. Language Model Prompts

Retrieving information from GPT-3 involves prompting the model with a few examples that serve as a few-shot learning training data. The content and the structure of the responses depend on these prompts. Therefore, experimentation with different prompts to achieve the desired structure is necessary. Formally, the examples that define the structure and content of an interaction with a language model are called *prompts*.

The purpose of a prompt is to 'show' the model how to interpret and respond to an input text appended to the prompt, which in our case is a question. For example, one wants to retrieve a piece of information about the most common items found on a table in a conference room. In such a case, the following prompt is constructed and used:

prompt:	*Q: What can be found on table in bar? Name five.* *A: bottle, class, cup, napkin, fork.* *Q: What can be found on table in conference room? Name five.*
GPT-3 response:	**paper, glass, laptop, phone, box.**

This example is a simple explanation of the role of the prompt. As it can be seen, the first part of the prompt—*Q* and *A*—is one-shot training data and 'teaches' the model that for a type of question like *Q*, a proper response looks like *A*. After that, the 'real' question *Q: What can be found on table in conference room?* is asked. Then, finally, the model responds with five items it 'thinks' represents the most suitable response.

Sometimes, one example is not enough, and multiple examples need to be provided to serve as few-shot learning training data. Empirically, we find that explaining the task and a well-defined question format help the model respond better.

To achieve more accurate results, we also utilize the chain-of-thought prompting method introduced in [20] for the fuzzy and the predicate expansion cases, described in Sections 5.2 and 5.3, respectively. In each example answer in the prompt, we hand-craft a reasoning that can help narrow down to the correct response. The model learns to generate a similar pattern and, as a result, generates a reasoning before answering the asked question.

5. Expansion of Commonsense Graph

To illustrate the benefits of using a language model for expanding the *WpKG*, we extract information from GPT-3 to construct different triples. It shows how versatile the interaction with the model can be and how different results are obtained. The presented utilization of GPT-3 involves the following scenarios:

- Asking for subjects and objects for given relations using a basic prompt template;
- Asking for the most and least likely subjects and objects for given relations to construct fuzzy triples;
- Asking for the most and least likely objects with novel relations given by a user.

Expanding the existing graph means 'asking' the language model to provide answers that contain the most suitable pieces of information that are directly added to the graph as nodes—subjects and objects—and relations that link the existing nodes to the newly added ones.

The questions are prepared based on templates that are initialized with facts/information obtained from the *WpKG* or from a user. Three sets of templates are constructed, one for each type of defined-above scenarios.

5.1. Simple Triples

In the beginning, a straightforward scenario that involves adding simple triples, i.e., triples that are not associated with degrees of strength of relations between subjects and objects, is presented. In such a case, GPT-3 is asked questions that result in retrieving from the model facts that are interpreted as subjects or as objects. It means that the questions are of the format $\langle ?s, relation_X, object_X \rangle$ when subjects are asked for, or $\langle subject_X, relation_X, ?o \rangle$ when objects are asked for. The retrieved subjects and objects are added as triples with the $relation_X$ to the *WpKG*.

In a nutshell, the process—for a single $relation_X$—is as follows:

- Extract five triples with the $relation_X$ from the *WpKG*.
- Select randomly one triple from the set of five, say, triple k; it is used in the process of customization of a prompt template for the $relation_X$.

  - Extract a set of five most popular objects Obj_k fitting $\langle subject_k, relation_X, \text{-}\rangle$ from the *WpKG*.
  - Extract a set of five most popular subjects Sub_k fitting $\langle \text{ -}, relation_X, object_k\rangle$ from the *WpKG*.
  - Audit the instantiated prompt and make changes if necessary.
- For each extracted triple $\langle subject_i, relation_X, object_i\rangle$:
  - Put $subject_i$ and $relation_X$ into the question template and append to the prompt.
  - Put the prompt to the language model to initiate the text generation.
  - Extract the five new objects Obj_{LM} from the generated text.
  - Add five new triples $\langle subject_i, relation_X, \text{-}\rangle$ with objects from Obj_{LM} to *WpKG*.
  - Put $relation_X$ and $object_i$ into the question template and append to the prompt.
  - Put the prompt to the language model to initiate the text generation.
  - Extract the five new subjects Sub_{LM} from the generated text.
  - Add five new triples $\langle \text{-}, relation_i, object_X\rangle$ with subjects from Sub_{LM} to *WpKG*.

As it is described above, the process of asking GPT-3 involves the instantiation of prompt templates. For the simple triples case, the prompt templates for asking for both *objects* and *subjects* are shown in Table 1. Following the aforementioned process, it can be seen that the prompts are filled out with facts/information obtained originally from *WpKG*, and the same initialization is used for prompting GPT-3 for all other *objects* or *subjects* obtained from the randomly selected $relation_X$s. Depending on the predicate $relation_X$, different variations of the prompt templates are created to result in meaningful questions and answers.

Table 1. Sample template for simple triple.

	SIMPLE_TEMPLATE_A for ⟨subject⟩
prompt:	*Answer with five items separated with comma.* *Q: What is* $\langle relation_X\rangle$ $\langle object_k\rangle$*? Name five.* *A: elements of* Sub_k *Q: What is* $\langle relation_X\rangle$ $\langle object_i\rangle$*? Name five.*
	SIMPLE_TEMPLATE_B for ⟨object⟩
prompt:	*Answer with five items separated with comma.* *Q: What* $\langle subject_k\rangle$ *can be* $\langle relation_X\rangle$*? Name five.* *A: elements of* Obj_k *Q: What* $\langle subject_i\rangle$ *can be* $\langle relation_X\rangle$*? Name five.*

The templates from Table 1 are used with five different relations: *behind, in, has, on,* and *watching*. The instantiated prompt templates, together with the results of querying GPT-3 for the relation *on*, are shown in Table 2 for extracting *subjects*, and in Table 3 for extracting *objects*.

It can be seen that, for example, selecting $object_A = plate$, we obtain the following triples: ⟨*food, on, plate*⟩, ⟨*drink, on, plate*⟩, ⟨*utensils, on, plate*⟩, ⟨*napkin, on, plate*⟩, and ⟨*tablecloth, on, plate*⟩, Table 2. Similarly selecting $subject_A = hair$, we obtain the triples such as ⟨*hair, on, head*⟩, ⟨*hair, on, beard*⟩, ⟨*hair, on, eyebrows*⟩, ⟨*hair, on, eyelashes*⟩, and ⟨*hair, on, pubic*⟩ (Table 3). Another example, this time in a graphical form, that shows an expansion of the triple ⟨*window, on, building*⟩ is illustrated in Figure 3. Besides the original triple, the figure includes its extension on both *subject* and *object* sides.

Table 2. Query and results for ⟨-, *on*, -⟩ for *subject*.

user:		*Answer with five items separated with comma.* *Q: What is on building? Name five.* *A: letter, door, sign, leaf, light.* *Q: What is* $\langle relation_A \rangle$ $\langle object_A \rangle$*? Name five.*
where:		$relation_A = on$ $object_A = \{ building, sign, man, plate, head \}$
GPT-3 responses for		
	building:	$subject_A \in \{letter, door, sign, leaf, light\}$
	sign:	$subject_A \in \{words, letters, numbers, shapes, colors\}$
	man:	$subject_A \in \{shirt, pants, belt, shoes, socks\}$
	plate:	$subject_A \in \{food, drink, utensils, napkin, tablecloth\}$
	head:	$subject_A \in \{hair, hat, ear, eyebrow, eyelash\}$

Table 3. Query and results for ⟨-, *on*, -⟩ for *object*.

user:		*Answer with five items separated with comma.* *Q: What window can be on? Name five.* *A: pole, car, bus, house, tree.* *Q: What* $\langle subject_B \rangle$ *can be* $\langle relation_B \rangle$*? Name five.*
where:		$subject_B = \{ window, letter, hat, food, hair \}$ $relation_B = on$
GPT-3 responses for		
	window:	$object_B \in \{pole, car, bus, house, tree\}$
	letter:	$object_B \in \{A, B, C, D, E\}$
	hat:	$object_B \in \{baseball, cowboy, graduation, party, winter\}$
	food:	$object_B \in \{apple, banana, orange, pear, grape\}$
	hair:	$object_B \in \{head, beard, eyebrows, eyelashes, pubic\}$

Of course, not all obtained *subjects* and *objects* are correct, especially in the case of asking for *objects*. For example, triples generated for the subject *letter*, Table 3, are quite inferior. A human-wise evaluation was performed; see Section 6.3 for details.

In the prompts, we chose the *What* question word, as it is generic enough to result in diverse types of results. However, a more fine-tuned selection of the question word may result in more relevant results, as suggested in [23].

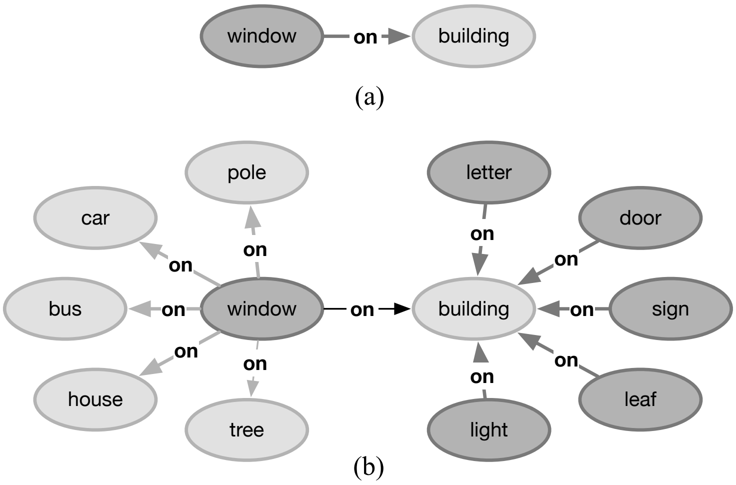

Figure 3. Expanded *WpKG*—simple triples: original triple (**a**); and after its extension (**b**).

We utilized the largest GPT-3 model, with 175 billion parameters, for the experiments. We started with a softmax temperature of 0.0 to obtain more deterministic results. However, we observed that the model sometimes shies away from generating text with this temperature setting and immediately generates an end token. To fix the problem, we increased the temperature to 0.7 and then to 1.0 to increase the chances for a good response.

5.2. Fuzzy Triples with Linguistic Terms

The remarkable abilities of GPT-3 can be utilized to extract *subjects* and *objects* when the triples need to be labeled with the degrees of the plausibility of their occurrence. Triples with such information can be added to the *WpKG* when the prompt, and its question-and-answer parts, used to query GPT-3 are constructed/designed in a specific way. The prompt templates presented in the previous section have to be modified.

To invoke responses from GPT-3 that give a quantifiable assessment of relation strength, the prompts should be more verbal to contextualize interaction with the model. The experiments with multiple approaches have led to the prompts that are the same, even if GPT-3 is asked to provide facts related to a variety of topics.

Due to the fact that two degrees of relation strength are considered, two prompts are designed and used: one for generating triples that represent high likeliness and one for building triples that are of low likeliness. Both of them are shown in Table 4. A quick look at them indicates that the prompts refer to quite different domains/topics—the questions are related to window and number. Yet, they work very well with the relations we use as examples—the same as for the simple triples in Section 5.1.

Another interesting 'feature' of these prompts is the very little need for instantiation. Only the last questions, Q_S for *subjects* and Q_O for *objects*, Table 4, are initialized to reflect the relations of interest.

Table 4. Template for fuzzy triple with linguistic terms.

	FUZZY_TEMPLATE_X for the linguistic term most likely
prompt:	*Answer with five items separated with comma.* *Q: What* **most likely** *has window? Name five.* *A: Window is usually used to see through. Therefore, train, building, house, car, bus.* *Q: What number can* **most likely** *be on? Name five.* *A: Number is made of digits and can be written on different things for information. Therefore, train, sidewalk, track, street, building.* Q_S: *What is* **most likely**$\langle relation_X \rangle$ $\langle object_X \rangle$*? Name five.* Q_O: *What does/is* $\langle subject_X \rangle$ **most likely** *be/-* $\langle relation_X \rangle$*? Name five.*
	FUZZY_TEMPLATE_Y for the linguistic term less likely
prompt:	*Answer with five items separated with comma.* *Q: What* **less likely** *has window? Name five.* *A: Window is usually used to see through. Therefore, hat, drawer, vase, basket, box.* *Q: What number can* **less likely** *be on? Name five.* *A: Number is made of digits and can be written on different things for information. Therefore, window, people, rock, tree, jacket.* Q_S: *What is* **less likely**$\langle relation_Y \rangle$ $\langle object_Y \rangle$*? Name five.* Q_O: *What does/is* $\langle subject_Y \rangle$ **less likely** *be/-* $\langle relation_Y \rangle$*? Name five.*

As an example of using the prompt templates, the results for a $relation_X = relation_Y = on$ are included. Please note that different question templates are developed to fit various types of relations. The obtained *subjects* and *objects* are in Tables 5 and 6 for the linguistic terms **most likely** and **less likely**, respectively.

Again, not all obtained *subjects* and *objects* are correct. For example, triples ⟨*hat, (***most likely***) on, -*⟩, Table 5, or ⟨*hat, (***less likely***) on, person*⟩, ⟨*food, (***less likely***) on, stove*⟩, Table 6, are quite inferior. As before, there is also a graphical representation in Figure 4 of the addition of new triples with the relation *on* that have *building* as their *object*. It can be seen that the most likely *subjects* are quite reasonable, while the less likely *subjects* are a bit odd. A human-wise evaluation is performed; see Section 6.3 for details.

For the **most likely** case, the softmax temperature starts at 0.0 and increases to 0.7 and 1.0, in the case that no text is generated. For the **less likely** case, we observe better results if the initial temperature is set to 0.7 and increases to 1.0 if needed.

Table 5. Query and results for ⟨-, **most likely** *on*, -⟩ for *object*.

user:		*Answer with five items separated with comma.* *Q: What* **most likely** *has window? Name five.* *A: Window is usually used to see through.* *Therefore, train, building, house, car, bus.* *Q: What number can* **most likely** *be on? Name five.* *A: Number is made of digits and can be written on* *different things for information.* *Therefore, train, sidewalk, track, street, building.* Q_O*: What* ⟨$subject_X$⟩ *can* **most likely** *be* ⟨$relation_X$⟩*? Name five.*
where:		$subject_X$ = *{ window, letter, hat, food, hair }* $relation_X$ = *on*
GPT-3 responses for		
	window:	$object_X \in$ *{train, building, house, car, bus}*
	letter:	$object_X \in$ *{train, sidewalk, track, street, building}*
	hat:	$object_X \in$ *{baseball cap, fedora, beanie, cowboy hat, sun hat}*
	food:	$object_X \in$ *{apple, banana, orange, grape, strawberry}*
	hair:	$object_X \in$*{person, animal, doll, toy, statue}*

Table 6. Query and results for ⟨-, **less likely** *on*, -⟩ for *object*.

user:		*Answer with five items separated with comma.* *Q: What* **less likely** *has window? Name five.* *A: Window is usually used to see through.* *Therefore, train, building, house, car, bus.* *Q: What number can* **less likely** *be on? Name five.* *A: Number is made of digits and can be written on* *different things for information.* *Therefore, train, sidewalk, track, street, building.* Q_O*: What* ⟨$subject_Y$⟩ *can* **less likely** *be* ⟨$relation_Y$⟩*? Name five.*
where:		$subject_Y$ = *{ window, letter, hat, food, hair }* $relation_Y$ = *on*
GPT-3 responses		
	window:	$object_Y \in$ *{number, people, rock, tree, jacket}*
	letter:	$object_Y \in$ *{number, people, rock, tree, jacket}*
	hat:	$object_Y \in$ *{window, book, cat, person, wall}*
	food:	$object_Y \in$ *{sink, counter, stove, refrigerator, table}*
	hair :	$object_Y \in$ *{shoulder, leg, foot, arm, hand}*

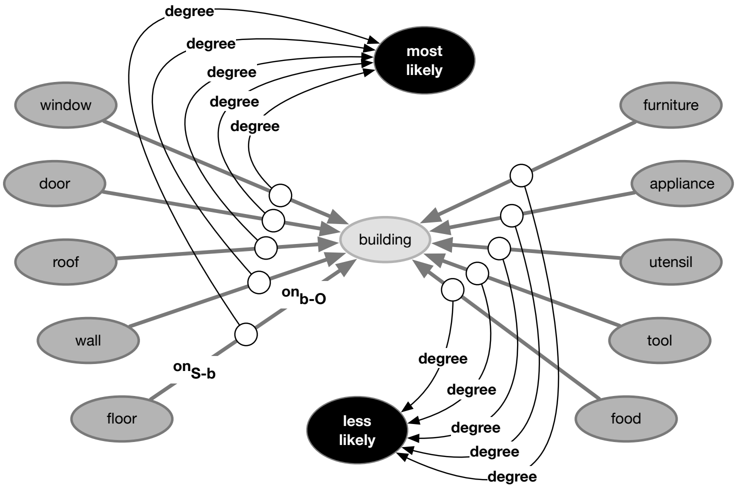

Figure 4. Expanded WpKG—triples with linguistic terms.

5.3. Fuzzy Triples with Novel User-Provided Relations

The last scenario focuses on the generation of new triples that contain novel relations provided by a user. It means the user gives relations that do not exist in the initial vision-based knowledge graph. We selected three novel relations: *used for*, *made of*, and *has property*. We opted for *triples with linguistic terms* and their respective prompts instead of the *simple triples* scenario, as more information about triples is obtained. The prompt templates used here are included in Table 4.

The results obtained for a $subject_X = arm$ and the user provided $relation_X \in$ *{used for, made of, has property}* are included in Table 7 for the fuzzy term **most likely**, and in Table 8 for the fuzzy term **less likely**. Graphically, the generated triples for $subject_X = shoe$ are in Figure 5. As in the previous cases, not all triples—constructed based on the obtained sets of objects—are satisfactory. The human evaluation results are presented in Section 6.3.

Table 7. Query and results for ⟨-, (**most likely**) *used for/made of/has property*, -⟩ for *object*.

user:	*Answer with five items separated with comma.* *Q: What* **most likely** *has window? Name five.* *A: Window is usually used to see through.* *Therefore, train, building, house, car, bus.* *Q: What number can* **most likely** *be on? Name five.* *A: Number is made of digits and can be written on* *different things for information.* *Therefore, train, sidewalk, track, street, building.* Q_O*: What is* $\langle subject_X \rangle$ **most likely** $\langle relation_X \rangle$*? Name five.*
where:	$subject_X = arm$ $relation_X \in$ **{used for, made of, has property}**
GPT-3 responses for	
used for:	$object_X \in$ *{lifting, carrying, pushing, pulling, holding}*
made of:	$object_X \in$ *{human, animal, plastic, metal, wood}*
has property:	$object_X \in$ *{to move, to bend, to be strong, to be flexible, to grip}*

Table 8. Query and results for ⟨-, (**less likely**) *used for*, -⟩ for *object*.

user:		*Answer with five items separated with comma.* *Q: What* **less likely** *has window? Name five.* *A: Window is usually used to see through.* *Therefore, train, building, house, car, bus.* *Q: What number can* **less likely** *be on? Name five.* *A: Number is made of digits and can be written on* *different things for information.* *Therefore, train, sidewalk, track, street, building.* Q_O*: What is* $\langle subject_Y \rangle$ **less likely** $\langle relation_Y \rangle$*? Name five.*
where:		$subject_X$ *= arm* $relation_X \in$ **{used for, made of, has property}**
GPT-3 responses for		
	used for:	$object_X \in$ *{hat, drawer, vase, basket, box}*
	made of:	$object_X \in$ *{metal, plastic, glass, wood, fabric}*
	has property:	$object_X \in$ *{number, window, glass, bottle, box}*

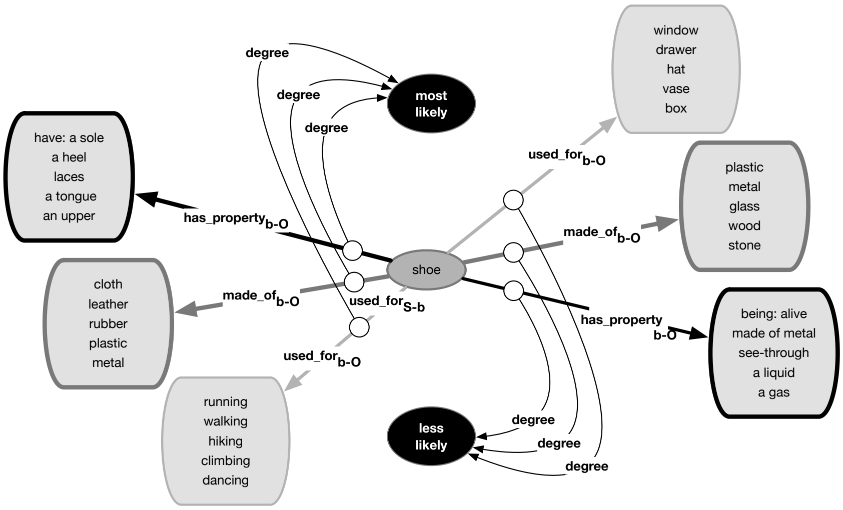

Figure 5. Expanded *WpKG*–fuzzy triples with *shoe* as their *subject* and user-provided relations *has_property, made_of, used_for*.

6. Discussion

The presented method for expanding existing commonsense knowledge graphs represents an example of a new approach to constructing knowledge graphs in a specific domain using very large language models and prompts. It can be said that these techniques are in their infancy; therefore, there are a number of aspects that need to be investigated regarding the approach itself as well as evaluation of the obtained results.

6.1. Vision-Based Commonsense Graph

Similar to how toddlers learn about their environment, our approach is based on two steps. First, we generate commonsense knowledge using vision models and then expand it using language models.

The evaluation of the weighted commonsense knowledge graph generated using only visual data is presented in Table 9 from our previous work [7,8]. Three different approaches for determining the weights (strengths) of relations are proposed and evaluated. Depending on the weighting mechanism, the accuracy of the generated commonsense triples ranges from 87.6% to 93%. Among these, the DPbM (detection probability-based method) correlates highly with human commonsense, while other methods still show good results.

Table 9. Human evaluation of the three weighting mechanisms defined in [8]. Three reviewers were given top 100 triples from each restaurant and classroom contextual commonsense knowledge graphs (total of 600 evaluations per method). Alpha is Krippendorff's Alpha [32] measuring consensus among evaluators.

Weighing Schema	**Accept**	**Reject**	**N/A**	**Accuracy (%)**	**Alpha**
DPbM	560	22	18	**93.0**	**0.78**
ROM	526	60	14	87.6	0.63
WOM	538	51	11	89.7	0.72

6.2. Preliminary Experiments with Language Models

The high accuracy obtained using automatic vision-based weighted commonsense knowledge generation does come with some specific challenges of its own. For example, the concept and relation vocabulary is limited only to the dictionary provided to the underlying models during the supervised training of the vision models. Adding a new vocabulary requires several time-consuming and costly tasks. They include human annotation on images to label objects and relations between them and then the fine-tuning of models for object detection and scene graph generation. Even if we accept the time and cost of adding a new vocabulary, it is shown in [9] that there is a bias toward the most common relationship type. It prevents the process from effectively going beyond specific vocabulary.

To address the issue of limited vocabulary, we have investigated using language models to extend the initial vision-based commonsense knowledge graph. We opted to use very large language models, such as GPT-3, for two main reasons. One is their capability to offer new concepts beyond the known ones with acceptable precision. The other reason is the flexibility and time/cost saving of using prompts instead of fine-tuning, which usually requires large amounts of costly human-annotated data.

Our experimental results support the overfitting statement explained in [24,25] stating that training on specific data reduces performance on novel data. We initially experimented with comparing one-shot-prompted 175-billion-parameter unsupervised-trained GPT-3 versus variations of smaller language models fine-tuned on an initial 5000-triple vision-based commonsense knowledge graph. Although the GPT-3 result accuracy was lower than a fine-tuned language model, the novelty of the vocabulary offered was much better. GPT-3 with 175 billion parameters predicted 15 times more vocabulary than the RoBERTa-large model with 355 million parameters.

6.3. Evaluation of Commonsense Knowledge Graph

To the best of our knowledge, there is limited benchmark data or a well-established method suitable for evaluating constructed commonsense knowledge graphs, especially when there are mostly novel generated concepts. There are benchmarks introduced in works such as [33], but are more related to knowledge base completion rather than expansion to new concepts. For mostly novel concepts, human evaluation of the results seems to be the preferred method, mainly in generative model scenarios, as performed in [2].

In this work, the process applied to assess the quality of the constructed commonsense knowledge graph is fully based on human evaluation using Amazon MTurk annotators. Amazon Mechanical Turk https://www.mturk.com (accessed on 12 August 2022) (MTurk) is a crowd-sourcing marketplace that provides, among multiple services, assistance in data annotation tasks. Three sets of validation tasks are performed for simple triples (Section 5.1), fuzzy triples (Section 5.2), and fuzzy triples with user-provided relations (Section 5.3).

The evaluation results are shown in Table 10 for only the new triples that did not exist in the original commonsense knowledge graph. As it can be seen, the results are encouraging. To gain some insight into the evaluation process and to better understand the evaluation results, it should be stressed that MTurk controls who is involved in the evaluation task. To increase the confidence in results, each triple is evaluated by three independent annotators.

To make the evaluation task easier and more intuitive for the annotators, we generated sentences from triples. Based on each predicate, a manual pattern is introduced. Once a sentence is generated using a fixed pattern, it is passed through an off-the-shelf grammar correction module to fix obvious errors. The sentences are then manually vetted to make sure they are grammatically correct and are based on the original triples.

In the description given, the annotators were asked to assume visual commonsense when encountering any of these statements. For example, in the case of *It is likely to see cloud behind cow.*, we asked them to imagine that they are in a field and they see cows. Then it makes sense to see clouds behind the cows.

Some examples of the triples and their evaluation scores are presented:

- *Shoe is used for running.* –> Correct with 0.95 confidence.
- *Shoe is not likely to be alive.* –> Incorrect with 0.95 confidence.
- *Shoe is not usually made of stone.* –> Correct with 0.65 confidence.

As we can see in the examples, finding a well-understood and easy-to-annotate verbalization of triples can affect the result. For example, in the case of *Shoe is not likely to be alive.*, the statement makes sense based on our understanding; however, it was not the case with the three annotators.

Table 10. Results of human evaluation of generated triples. Overall, *Likely* and *Unlikely* columns show the accuracies regarding total triples, most-likely triples, and less-likely triples, respectively. *N* represents the number of triples evaluated in each case.

Triple Type	***N***	**Overall Accuracy**	***Likely* Accuracy**	***Unlikely* Accuracy**
Simple	122	72.95%	N/A	N/A
with Linguistic Terms	287	67.94%	68.09%	67.81%
with New Relations	148	72.97%	66.22%	79.73%

A few examples are analyzed under Table 11 to understand the obtained results better. Triples without linguistic terms are called *Simple*. Triples *with Linguistic Terms* contain two terms, **most likely** and **less likely**. Triples *with New Relations* refer to triples with linguistic terms generated with predicates that do not exist in the initial commonsense knowledge graph. For brevity, the initial parts of the prompts are removed. Only the last part of the prompt (question) is kept. The process of generating triples *with Linguistic Terms* and *with New Relation* uses the chain-of-thought prompting methods, shown in Sections 5.2 and 5.3, while *Simple* triples are generated using a simple question and answering prompting method, shown in Section 5.1.

The obtained results are compared with the results found in similar works. TransOMCS paper [34] reports an overall accuracy of 56% while focusing on the automatic mining of commonsense knowledge from linguistic graphs. The results in TransOMCS are based on 100 randomly selected tuples from the overall results set, which five Amazon mTurk workers evaluated. Another comparable work focuses on symbolic knowledge distillation from large language models, mostly about commonsense social relations, without relationship weights [2]. This work reports a human-evaluated correctness percentage of 73.3% when GPT-3 is used with prompts to complete a knowledge graph. The reported value is close to the comparable case of *Simple* triples as shown in Table 10. The approach used in [2] requires text completion for every subject and predicate to generate each triple. On the other hand, our approach uses prompts that generate $N = 5$ new concepts during a single run. It results in roughly one-fifth of the cost when both methods use the same model.

Table 11. Examples: two correct and one incorrect for each type of generated triple. Correct parts of the response are in teal color, while the incorrect parts are in red color.

Simple	with Linguistic Terms	with New Relations
Q: What is in room? Name five. A: bed, table, chair, lamp, computer.	Q: What is most likely to be in bowl? Name five. A: Bowl is a container and can hold different things. Therefore, rice, cereal, milk, soup, fruit.	Q: What is shoe most likely used for? Name five. A: Shoe is used to protect feet. Therefore, running, walking, hiking, climbing, dancing.
Q: What does face have? Name five. A: eyes, nose, mouth, ears, chin.	Q: Where is flower less likely in? Name five. A: Flower is a natural thing and needs soil, water and sunlight to grow. Therefore, sink, oven, book, pencil, phone.	Q: What property does sidewalk most likely have? Name five. A: Sidewalk is a path for pedestrians. Therefore, it is usually level, made of concrete, has a curb, is smooth, and is wide.
Q: What food can be on? Name five. A: apple, banana, orange, pear, grape.	Q: What hat can most likely be on? Name five. A: Hat is worn on the head and can be made of different materials. Therefore, baseball cap, fedora, beanie, cowboy hat, sun hat.	Q: What property does arm less likely have? Name five. A: Arm is a body part and usually has skin, muscle, bone, and hair. Therefore, number, window, glass, bottle, box.

To further demonstrate the scalability of the proposed method, we generated 1905 triples with linguistic terms. Triples with 13 different predicate types from our vision-based commonsense knowledge graph were used for the generation purpose. There are 1075 triples with the linguistic term **less likely** and 830 with the term **more likely**.

All the triples were evaluated using three Amazon mTurk annotators on the Amazon SageMaker platform. The human evaluations of **more likely** triples resulted in higher accuracy of 72.15%, while the **less likely** triples resulted in an accuracy of 62.1%. We only considered triples with at least 95% evaluation confidence among the three annotators (662 triples). The evaluation of triples with different predicate types and linguistic terms resulted in different accuracies, as shown in Figure 6. This scaling experiment shows that the generated dataset size can expand from the initial hundreds of triples to thousands and beyond.

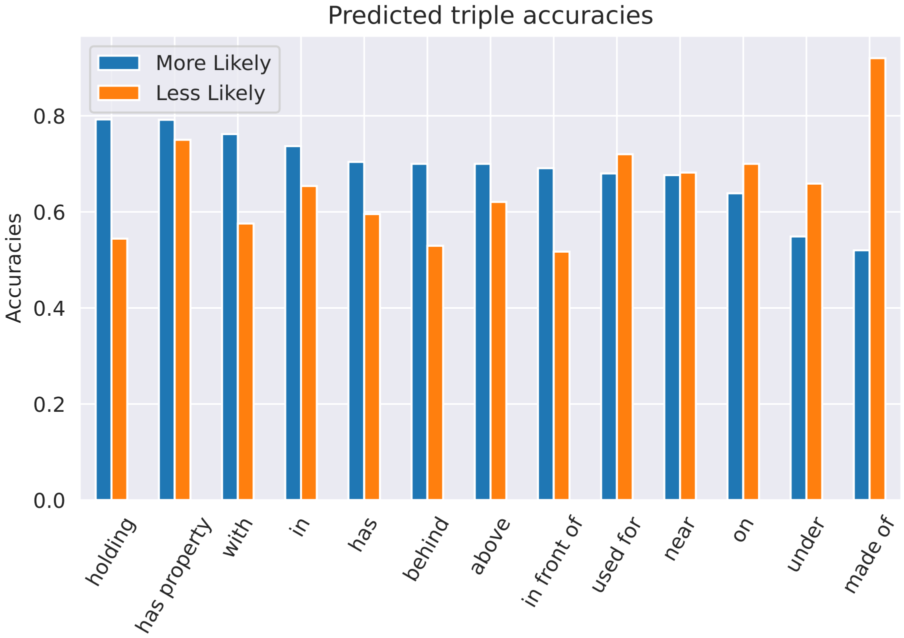

Figure 6. Human (mTurk) annotation accuracy of different predicate types and linguistic terms.

7. Conclusions

There is a growing interest and a need for collecting and storing knowledge that represents information about real-world scenarios and things and activities of everyday life. That type of information—named commonsense—becomes essential when one wants to build autonomous systems that exist around us and assist us in daily duties.

The commonsense knowledge is present in different visual and verbal forms and is learned via observations, experiences, and interaction with others.

A simple attempt to address extracting commonsense knowledge and representing it as a graph is presented here. The previous work [7] showed a method of analyzing images and constructing a commonsense knowledge graph via the fusion of multiple scene graphs extracted from images.

This paper, perceived as a continuation of the work on images, presents a methodology for expanding existing commonsense graphs with facts retrieved from language models. The development of very large language models opens an opportunity to use them for multiple tasks involving retrieving pieces of information and facts in various domains. This capability of the models was utilized here to pull out commonsense information that is easily added to the existing knowledge graphs. Specific prompts and their templates were constructed to retrieve related information. This information was transformed into triples and added to the commonsense graph. Three different types of new triples were considered: simple ones, fuzzy ones with linguistic terms describing degrees of their likeliness, and ones with specific relations provided by the user.

A validation process of new triples was designed and executed—the Amazon service called Mechanical Turk was utilized. The obtained evaluations confirmed the usefulness of the proposed methodology for expanding commonsense graphs.

At the same time, more work is needed to construct prompts that improve the correctness of retrieved information and create triples with more subtle degrees of likeliness. Additionally, more investigation regarding the suitability of different language models is mandated. In this paper, we used the chain-of-thought prompting method [20]. While this prompting method leads to good results, it seems interesting and important to investigate other prompt methods, such as [35], to see if better and more accurate results are achievable.

Author Contributions: Conceptualization, N.R. and M.Z.R.; methodology, N.R. and M.Z.R.; software, N.R. and M.Z.R.; validation, N.R. and M.Z.R.; formal analysis, N.R. and M.Z.R.; investigation, N.R. and M.Z.R.; resources, N.R. and M.Z.R.; data curation, N.R. and M.Z.R.; writing—original draft preparation, N.R. and M.Z.R.; writing—review and editing, N.R. and M.Z.R.; visualization, N.R. and M.Z.R.; supervision, M.Z.R.; project administration, N.R. and M.Z.R.; funding acquisition, M.Z.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Autonomous Systems Initiative (ASI), Alberta, Canada.

Data Availability Statement: The code, data, and experiments are going to be available under the following repository: https://github.com/navidre/lm_vision_kg_expansion (accessed on 10 August 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this paper:

WpKG	World-Perceiving Knowledge Graph
C-WpKG	Contextual World-Perceiving Knowledge Graph
GPT	Generative Pre-Trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
AI	Artificial Intelligence
DPbM	Detection Probability-Based Method
ROM	Relative Occurrence Method
WOM	Weighted Occurrence Method
VG	Visual Genome

References

1. Hwang, J.D.; Bhagavatula, C.; Bras, R.L.; Da, J.; Sakaguchi, K.; Bosselut, A.; Choi, Y. COMET-ATOMIC 2020: On Symbolic and Neural Commonsense Knowledge Graphs. In Proceedings of the AAAI, Virtual Conference, 2–9 February 2021.
2. West, P.; Bhagavatula, C.; Hessel, J.; Hwang, J.D.; Jiang, L.; Bras, R.L.; Lu, X.; Welleck, S.; Choi, Y. Symbolic knowledge distillation: From general language models to commonsense models. *arXiv* **2021**, arXiv:2110.07178.
3. LeCun, Y. A Path Towards Autonomous Machine Intelligence Version 0.9.2. Available online: https://openreview.net/pdf?id=BZ5a1r-kVsf (accessed on 27 June 2022).
4. Choi, Y. The Curious Case of Commonsense Intelligence. *J. Am. Acad. Arts Sci.* **2022**, *151*, 139–155. [CrossRef]
5. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the opportunities and risks of foundation models. *arXiv* **2021**, arXiv:2108.07258.
6. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the EMNLP, Austin, TX, USA, 1–5 November 2016.
7. Rezaei, N.; Reformat, M.Z.; Yager, R.R. Image-Based World-perceiving Knowledge Graph (WpKG) with Imprecision. *Inf. Process. Manag. Uncertain Knowl. Based Syst.* **2020**, *1237*, 415–428.
8. Rezaei, N.; Reformat, M.Z.; Yager, R.R. Generating Contextual Weighted Commonsense Knowledge Graphs. In Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Milan, Italy, 11–15 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 593–606.
9. Tang, K.; Niu, Y.; Huang, J.; Shi, J.; Zhang, H. Unbiased Scene Graph Generation From Biased Training. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3713–3722.
10. McCarthy, J. *Formalizing Common Sense*; Intellect Books: Bristol, UK, 1990; Volume 5.
11. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of the Advances in Neural Information Processing Systems*; Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; Volume 26.
12. Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings; Bengio, Y., LeCun, Y., Eds., 2015.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
14. Wang, C.; Liu, X.; Song, D.X. Language Models are Open Knowledge Graphs. *arXiv* **2020**, arXiv:2010.11967.

15. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language Models as Knowledge Bases? 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), Hong Kong, China, 3–7 November 2019; pp. 2463–2473.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers); Association for Computational Linguistics: Minneapolis, SP, USA, 2019; pp. 4171–4186. [CrossRef]
17. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
18. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X.V.; et al. Opt: Open pre-trained transformer language models. *arXiv* **2022**, arXiv:2205.01068.
19. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *arXiv* **2022**, arXiv:2204.02311.
20. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; Zhou, D. Chain of thought prompting elicits reasoning in large language models. *arXiv* **2022**, arXiv:2201.11903.
21. Rezaei, N.; Reformat, M.Z. Super-Prompting: Utilizing Model-Independent Contextual Data to Reduce Data Annotation Required in Visual Commonsense Tasks. *arXiv* **2022**, arXiv:2204.11922.
22. Khot, T.; Sabharwal, A.; Clark, P. What's Missing: A Knowledge Gap Guided Approach for Multi-hop Question Answering. 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), Hong Kong, China, 3–7 November 2019; pp. 2814–2828.
23. Fabbri, A.R.; Ng, P.; Wang, Z.; Nallapati, R.; Xiang, B. Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 4508–4513.
24. Jastrzębski, S.; Bahdanau, D.; Hosseini, S.; Noukhovitch, M.; Bengio, Y.; Cheung, J. Commonsense mining as knowledge base completion? A study on the impact of novelty. In Proceedings of the Workshop on Generalization in the Age of Deep Learning, Munich, Germany, 8–14 September 2018; Association for Computational Linguistics: New Orleans, LO, USA, 2018; pp. 8–16. [CrossRef]
25. Davison, J.; Feldman, J.; Rush, A.M. Commonsense knowledge mining from pretrained models. 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), Hong Kong, China, 3–7 November 2019; pp. 1173–1178.
26. Chen, X.; Shrivastava, A.; Gupta, A. NEIL: Extracting Visual Knowledge from Web Data. In Proceedings of the IEEE International Conference on Computer Vision 2013, Sydney, Australia, 1–8 December 2013; pp. 1409–1416.
27. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4444–4451.
28. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef] [PubMed]
29. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
30. Zellers, R.; Yatskar, M.; Thomson, S.; Choi, Y. Neural Motifs: Scene Graph Parsing with Global Context. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5831–5840.
31. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The Curious Case of Neural Text Degeneration. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
32. Hayes, A.F.; Krippendorff, K. Answering the Call for a Standard Reliability Measure for Coding Data. *Commun. Methods Meas.* **2007**, *1*, 77–89. [CrossRef]
33. Li, X.; Taheri, A.; Tu, L.; Gimpel, K. Commonsense Knowledge Base Completion. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; (Volume 1: Long Papers); Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1445–1455. [CrossRef]
34. Zhang, H.; Khashabi, D.; Song, Y.; Roth, D. TransOMCS: From linguistic graphs to commonsense knowledge. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 4004–4010.
35. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv* **2022**, arXiv:2203.11171.