COMMONSENSE KNOWLEDGE OF PROTOTYPICAL FUNCTIONS FOR NATURAL LANGUAGE PROCESSING

by

Tianyu Jiang

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

School of Computing

The University of Utah

August 2023

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Tianyu Jiang**
has been approved by the following supervisory committee members:

Ellen M. Riloff ,	Chair(s)	**07/10/2023** Date Approved
Ana Marasović ,	Member	**07/10/2023** Date Approved
Jeffrey M. Phillips ,	Member	**07/10/2023** Date Approved
Nathan Schneider ,	Member	**07/18/2023** Date Approved
Vivek Srikumar ,	Member	**07/11/2023** Date Approved

by **Mary Hall** , Chair/Dean of
the Department/College/School of **Computing**
and by **Darryl P. Butt** , Dean of The Graduate School.

ABSTRACT

Commonsense knowledge has long been recognized as an inevitable source of information for natural language understanding. In this research, I focus on one specific type of commonsense knowledge that people use in everyday living: “functional knowledge.” People go to different places for a common set of goals: people go to schools to study, go to stores to buy clothing, and go to restaurants to eat. Comparably, people create and use physical objects for different purposes: knives are for cutting, cars are for transportation, and phones are for communication. I refer to the goals people go to a place to achieve or the purpose people use an object for as “prototypical functions” of locations or objects.

People often infer a richer meaning for sentences than what they “explicitly” state. I argue that the commonsense knowledge of prototypical functions is essential for natural language understanding systems to “read between the lines” and make the same types of inferences that humans do.

I will first introduce a semi-supervised learning technique to learn the prototypical functions of locations from large corpora and a transfer learning method with pretrained language models to learn the prototypical functions of human-made physical objects. After that, I will demonstrate how to use this prior knowledge of functions in downstream applications. Our human annotations results show that when an object is being used in a sentence, we can predict with high confidence that the object is used for its prototypical function. However, sentences often mention objects when they are not used at all. So I design an automatic system to recognize the true use of an object, in order to enable prototypical function inferences, especially when the action is implicit. I also show the importance of functional knowledge for computer vision. Situation recognition is a task to recognize the main activity depicted in a static image. I build a transformer-based model that incorporates functional knowledge of objects in the image to better identify the depicted activity for this task.

To my wife and my parents.

CONTENTS

CHAPTERS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

If going back in time, I would still choose this long journey. I am grateful to many people, without whom I will not be where I am.

First and foremost, I want to thank my advisor Ellen Riloff for her support through the years. Ellen introduced me to this fantasy world of Natural Language Processing research and always serves as a researcher role model whom I am still learning from till this day. I cannot imagine a better advisor with all her insight, energy and cheerful personality.

I would like to thank other members in my thesis committee: Vivek Srikumar, Jeff Phillips, Nathan Schneider and Ana Marasović, not only for their helpful comments and feedback on this thesis, but also for their guidance for me to view my work from different perspectives, and encouragement for me explore the fields that I didn't think of.

I am grateful to have met with many smart, humorous and creative people during the past few years: Haibo Ding, Xingyuan Pan, Jie Cao, Youngjun Kim, Tao Li, Yichu Zhou, Xiaowan Li, Yuan Zhuang, Zhenduo Wang, Vivek Gupta, Maitrey Mehta, Mattia Medina-Grespan, Ashim Gupta, Nate Stringham, Atreya Ghosal, Fateme Hashemi, Jacob Johnson, Xin Yu, Yufan Xu, Jinghua Yan, Nikita Bhutani, Yoshihiko Suhara, Estevam Hruschka, Ladislav Kavan, Al-Halah Ziad, and Shandian Zhe. A special thank goes to Yichu Zhou and Yuan Zhuang, for not just countless discussions and brain storming, but all the hours and effort we spent together at the gym — a healthy body is the foundation to pursue any intelligence. I cherish the memories with you all.

I am indebted to my friends and family for their support and love, especially my parents Dong Jiang and Xiu Zhu. They are my source of strengths.

Last but not least, I would like to thank my wife Yue Yang, without whom I cannot enjoy such a lovely voyage in Utah.

CHAPTER 1

INTRODUCTION

Commonsense knowledge plays an important role in our daily life. It has been widely acknowledged as understandings of the everyday world and has long been recognized as an inevitable source of information that a system should use for natural language understanding. A well-known example illustrating the importance of commonsense knowledge came from the Winograd Schema Challenge [85]:

```
The city councilmen refused the demonstrators a permit
        because they [feared/advocated] violence.
```

The choices of *feared* and *advocated* demand answering the underlying question "Does the pronoun 'they' refer to the city councilmen or the demonstrators?" The correct answer is obvious to the human readers due to our commonsense but proves difficult for automatic natural language processing systems.

Commonsense knowledge differs from encyclopedic knowledge, which is more factual and typically well defined in a textbook. Instead, commonsense is multifarious and "everything everywhere". For example, there is taxonomy commonsense knowledge so we know *an apple is a fruit* and *a dog is an animal*; there is temporal commonsense knowledge in our minds that *a football game takes longer than a UFC fight*, as well as *a toddler is too young to get a Ph.D. degree*; there is spatial commonsense knowledge that *a desk is often accompanied by chairs* so that when we see a still picture where a piece of wood is peeking above a table, we would know that it is part of a chair, rather than a random plank [40].

In this work, I will focus on one specific type of commonsense knowledge that people use in everyday living — **functional knowledge**. Specifically, I consider functional knowledge associated with two types of entities, *locations* and *human-made physical objects*.

1.1 Motivation

If you came across your friend who told you that he has just been to a restaurant, you could naturally ask "how was the food" because you assume that people go to restaurant to eat, unless your friend is a cook. We observe and hypothesize that it is a common fact that people go to different places to accomplish goals. People go to stores to buy clothing, go to libraries to study, and go to the doctor for medical services. For many places, people typically go there for a common set of reasons, which I will refer to as the **prototypical function (goal activities)** of a *location*.

The second type of entity, *human-made physical objects*, refer to those concrete things created by humans, as opposed to 1) things that occur naturally such as a rock, a river, or 2) abstract terms or ideas such as Marxism or the theory of relativity. A book is a human-made physical object (a paragraph is not), a desk is a human-made physical object, and so is a car. I intentionally focus on human-made objects as opposed to naturally occurring objects because they were typically created for a reason (note how you know what "they" refers to!), i.e., why people need this object – the "purpose". For example, people use a knife for cutting, pick up a book for reading, etc. Limiting the scope to physical objects is also because abstract creations by human beings often fall into the axis of specialized knowledge, which is opposite to the commonsense knowledge about everyday living. I refer to the most typical way people use a *physical object* as its **prototypical function**.

I am interested in the purpose of locations and objects as it is one type of commonsense knowledge that can be easily overlooked yet plays an important role in our daily language understanding. People may not realize how many assumptions they unconsciously make based on the common purpose of the objects when they deal with textual or speech information. Imagine a reading scenario, when you read

```
John put the chicken in the oven.
```

You would instantly know that John is cooking because you know ovens are used for heating food and chicken is edible. You would probably assume it is before lunch or dinner time and John is in the kitchen, where the oven is located. You may even know that the chicken will stay in the oven for an hour or so, neither one minute nor multiple days. None of this information is explicitly expressed in this simple sentence. However, humans are able to make these assumptions without any difficulty because our commonsense knowledge of chicken, oven and the unmentioned kitchen is unconsciously triggered.

This thesis was also motivated by observing sentences and realizing that we can infer the "true" meaning for these sentences beyond what they "explicitly" state. Oftentimes, true actions are left implicit in natural language. In many cases, these actions do not need to be explicitly stated because they can be easily inferred by people using our commonsense knowledge. Consider the following sentences:

```
(a) She went to the kitchen and got chicken.
(b) She went to the supermarket and got chicken.
(c) She went to the restaurant and got chicken.
```

Sentences (a) - (c) illustrate how we infer different actions based on the **locations**. In sentence (a), we infer that she retrieved chicken. In (b), we infer that she paid for the chicken. In (c), we infer that she also paid for the chicken, but more importantly, she ate the chicken. Note how the verb "got" maps to different presumed events depends on the location.

```
(d) He finished the cigarette.
(e) He finished the puzzle.
(f) He finished the movie.
```

Sentences (d) - (f) show how we infer different actions based on the **objects** when the main action is elided (i.e., "finished" means that some action has ended but the action itself is implicit). Most people would assume that the cigarette was smoked, the puzzle was solved, and the movie was watched.

This type of inference may seem trivial for a human being, yet still remains a difficult problem for an automatic system in the field of natural language processing.

There has been a considerable amount of work in the natural language processing (NLP) community to tackle the semantic understanding of natural language. One of the widely known NLP tasks that are closely related is the semantic role labeling task, which aims to assign labels to words or phrases in a sentence that indicates their semantic roles, in order to answer “who did what to whom, when, where and why, etc.” However, as it relies on the predicate-argument structure for parsing, its ability to make inference of the true meaning is limited by the explicit predicate (e.g., usually the root verb). Using the same example sentence above, “John put the chicken in the oven”, a typical semantic role labeling system would identify the verb “put” as the root and trigger a “physical placement” event, then John is the putter (agent), the chicken is the thing being put (theme), and the oven is the destination. It provides an understanding of the sentence based on the explicit words without any implicit but true meaning that human readers would achieve. Similar weaknesses can be found in a number of related tasks, such as frame semantic parsing [7, 53], in which the identified frame (event) can only be triggered if one of its lexical units exists in the sentence; abstract meaning representation [8], which also relies on the predicate-argument senses from the PropBank [115]; and so on.

I would argue that the commonsense knowledge of **prototypical functions** is essential for natural language understanding systems to “read between the lines” and make the same types of inferences that people do. For a NLP system to understand the implicit purpose behind an event as indicated in the previous examples, it needs extra knowledge associated with the object, the location, and the event. Specifically, this brings up three questions that unify this thesis:

1.1.1 How Should We Represent Functional Knowledge?

A key question to answer before building any knowledge-based system is how should we represent the knowledge. Some previous work on commonsense knowledge acquisition has opted to collect natural language words and phrases as expressions in a subject-object relation triple, such as ConceptNet [159]. Following this paradigm, I tackled the problem using natural language to describe the function of locations.

For physical objects, I explored using frames from FrameNet as a canonical representation for each type of general function that a physical object could have. A frame is a conceptual structure that usually characterizes an abstract scene or situation, which originates from the study of frame semantics [51]. The Berkeley FrameNet project [141] provides an online lexical database for frame semantics, which contains 1,221 frames and their semantic roles and lexical units. Compared with natural language phrases, a canonical representation naturally captures clusters of objects with the same function and avoids evaluation issues arising from differing phrases with the same meaning. Besides, FrameNet provides rich semantic structures to capture more information and can be adapted to other work that is also based on FrameNet.

1.1.2 How Can We Learn Functional Knowledge?

One challenge for learning commonsense knowledge from text is that people often do not explicitly mention it in speaking or writing. Some prior work has addressed this issue by using crowd-sourcing to collect commonsense knowledge. However, human annotation is expensive and it is difficult to imagine that all human knowledge will be covered by limited annotations. To tackle the problem, I explored 1) *semi-supervised learning* that uses a small amount of labeled data to infer the knowledge for the unlabeled data; 2) *transfer learning* using a large pre-trained language model and lexical databases to acquire the knowledge.

1.1.3 When and Where Should We Apply Functional Knowledge in Real-World Application?

It has long been an open research question of how artificial intelligence systems could benefit from existing commonsense knowledge. A big challenge for applying knowledge of functions is that physical objects are often mentioned when they are not used at all. So I design a system that identifies whether a physical object mentioned in a sentence has been/is being used or will likely be used in the future. The goal is to identify sentences that imply the true use of an object to enable inferences using the functional knowledge when the action is implicit. Apart from textual understanding, we believe that multimodal tasks that require different types of intelligence should also benefit from the commonsense knowledge. Specifically, as a downstream application as well as an extrinsic evaluation, we select a computer vision task, Situation Recognition, to demonstrate the value of our prior learned functional knowledge.

1.2 Research Contributions

The primary contributions of this dissertation can be summarized as follows:

Claim #1: Commonsense knowledge about the prototypical functions can be learned

> *from large corpora and language models.*

We first explored a semi-supervised statistical method that used a small set of labeled data as “seeds” to iteratively learn new knowledge from a large text corpus. The acquired knowledge achieved a better quality than several baseline methods for the defined task.

Inspired by the success of transfer learning using large pre-trained language models, we then explored using the language models in two different ways to predict the functional knowledge: 1) prompting with manually pre-defined templates to fill in the blanks; 2) encoding the dictionary definitions for determining semantic similarity. The experimental results show that by combining these two methods together, the joint model substantially outperforms several baselines, as well as an existing knowledge base, ConceptNet.

> *Claim #2: Commonsense knowledge about the prototypical functions can benefit downstream artificial intelligence applications.*

A big question for commonsense knowledge research is how to apply the knowledge in real practice. We explored two paths to tackle this question: 1) textual understanding and 2) visual recognition.

Our initial intuition for using the knowledge is to understand implicit sentences, e.g., ”he used a washcloth on the floor” should be represented as a *cleaning* event, not *using* event. But the challenge for applying our knowledge of functions is that physical objects are often mentioned when they are not used at all, e.g., the knife fell off the table. Motivated by this, I designed an automatic system to classify the usage status of a physical object mentioned in a sentence into three categories: *Used*, *Anticipated Use*, and *No Use*. The goal is to recognize the true use of an object to enable prototypical function inferences, especially when the action is implicit. Our annotation results demonstrate that when a sentence mentions or implies the use of an object, **96%** of these sentences correspond to the typical function of the object. This data suggests that if we had a perfect object use classifier, we could infer how an object was used with 96% accuracy simply by assuming its prototypical function. Our transformer-based model incorporates two types of data augmentation techniques. It substantially outperforms prompting-based methods with two well-known language models: GPT-2 [133] and T0++ [145].

It has been recognized that commonsense knowledge does not realize its full potential in many of the existing crossmodal systems [186]. It is worth investigating how to inject commonsense knowledge into a multimodal model through textual inputs. To start, we selected the situation recognition problem [191], which is the task of recognizing the activity depicted in an image, as a downstream application of our functional knowledge. We built an architecture that compares the performance of recognizing the activity with and without functional knowledge of objects. Our experimental results show that when injecting prototypical function frames, the system achieves better performance.

1.3 Dissertation Outline

To help the readers navigate through this dissertation, we describe the primary content of each chapter as follows:

- **Chapter 2:** Description of the existing work related to the commonsense knowledge, functional knowledge as well as other work related to commonsense knowledge of locations and objects.

- **Chapter 3:** Discussion of research on learning functional knowledge of locations. This chapter describes the definition of the *prototypical goal activities* for locations; a NLP task of identifying the prototypical goal activities; a human-annotated dataset that is used to evaluate the task; and a statistical model that automatically learns the functional knowledge from a large corpus.

- **Chapter 4:** Discussion of research on learning functional knowledge of human-made physical objects and FrameNet frames as its representation. This chapter describes the definition of the *prototypical functions* for human-made physical objects; a NLP task of identifying the prototypical functions of objects based on FrameNet frames; a human-annotated dataset that is used to evaluate the task; a frame identification system that is evaluated on standard frame semantic parsing datasets, which serves as a module of the final system; and a final system to learn the functional knowledge from a pre-trained language model.

- **Chapter 5:** Discussion of research on applying functional knowledge of objects to sentences. It is a common phenomenon that objects are often mentioned when they are not used at all, which poses a big challenge for applying the functional knowledge in textual understanding. This chapter describes the task of identifying the usage status of the objects mentioned in a sentence; a human-annotated dataset that is created to evaluate the task; and a transformer based system to classify the usage status into one of the three categories.

- **Chapter 6:** Application of functional knowledge of objects for a computer vision task. This chapter presents a model that use the prior functional knowledge of objects to help identify the main activity depicted in a still image. It is evaluated on a computer vision dataset created for situation recognition. The experiments demonstrate the effectiveness of the functional knowledge for multimodal scene understanding.
- **Chapter 7:** Conclusions and future work for this research topic.

CHAPTER 2

BACKGROUND

In this chapter, we will discuss previous work that relates to this dissertation. We will first take a look at the background of commonsense knowledge in the field of natural language processing, then focus on work related to functional knowledge, as well as commonsense knowledge of objects and locations.

2.1 Commonsense Knowledge in NLP

It has long been recognized that commonsense knowledge plays an important role in natural language processing [30, 183]. This observation has led to considerable work toward learning and applying commonsense knowledge for natural language understanding.

2.1.1 Early Attempts

In the early days of artificial intelligence, Conceptual Dependency theory was proposed by Roger Schank with the goal of developing a "representation of the conceptual base that underlies all natural languages" [147, 148]. It introduced a small set of primitive actions from which the representations can be built. For example, one primitive act ATRANS means "transfer of an abstract relationship," so that a sentence such as "John gave a book to Mary" is then represented as an ATRANS action with the *donor* (John), the *recipient* (Mary) and the *object* (book). The claim was that this small set of representational elements could be used to produce a canonical form representation for English sentences (and other natural languages) [96]. In close relation to my work, physical objects also play an important role in conceptual dependency. For example, in conceptual dependency [147, 148], some primitive acts like PTRANS and PROPEL are explicitly defined involving a physical object (in an actor-action-object framework).

Then starting from the 1970s, multiple knowledge organization theories descended from Conceptual Dependency were proposed. For example, Schank and Abelson [149] introduced a larger knowledge structures, called scripts, that "precompiled" sets of likely activities in a common situation. Scripts represented stereotypical sequences of events. A script consists of a set of roles involved in the script; and scenes that describe the typical events that are part of the script chain. For example, in the $RESTAURANT script, *roles* include the customer, the waiter, the cook, the restaurant, the food, etc. *Scenes* included ENTER, ORDER, EAT, PAY, and LEAVE. Then each scene is decomposed into a sequence of conceptual dependency representations. One influential follow-up work in this area, Chambers and Jurafsky [26], treats the problem of script induction as learning narrative event chains. They demonstrate that this type of knowledge can be learned unsupervised from raw text using textual co-occurrence statistics. A number of publications used statistical methods to learn the script knowledge [69, 124, 140]. McIntyre and Lapata [102] used script knowledge to generate short children's stories. More recently, neural network models have been proposed to learn the script knowledge [61, 197], including pre-trained language models [144].

In close relation to this dissertation, "frames" are also well-known structures to represent knowledge. Frame theory is a study of how we associate words and phrases with conceptual structures called frames [105]. A frame usually characterizes an abstract scene or situation. Frame theory asserts that people understand the meaning of words by virtue of the frames which they evoke [51]. For example, a *cooking* frame will describe a common food preparation scenario where a cook makes food from ingredients using some heating equipment. It may further describe how much time the cooking procedure takes, and the containers for the food. Frames can also be very simple actions. For example, a *placing* frame involves only an agent who does the placing, an object to be placed somewhere, and a place.

2.1.2 Commonsense Knowledge Resources

2.1.2.1 Expert Curated

Many of the existing commonsense knowledge bases are built manually by experts since the construction of curated knowledge bases typically leads to a high quality. Following the line of knowledge engineering with manually written facts about the world, Cyc [84] is one of the most well-known projects to capture commonsense knowledge. The Cyc Project [84] was started by Lenat in 1984. Cyc contains a Cyc knowledge base (Cyc KB) which encodes the commonsense knowledge in formal language CycL based on first-order logic. Millions of facts and rules were formally codified by ontologists skilled in CycL. Cyc also contains an inference engine, which was designed to perform general logical deduction, as well as other inference methods. Numerous other publications have advocated a logic-based approach to knowledge representation and reasoning [62, 64, 100, 109]. Previous work on qualitative spatial representation also proposed different theories of logic systems based on mathematical topology [34, 179].

Following the theory of frame semantics, the Berkeley FrameNet project [5, 141] provides an online lexical database for frame semantics and a corpus of annotated documents. FrameNet [141] lexical database consists of over 1,200 hierarchically-related semantic frames and 13,000 lexical units, was created by linguists of the Berkeley FrameNet project team. The FrameNet project also provides more than 200,000 annotated sentences. To achieve that, they designed an annotation editor software with a graphical user interface so that well-trained Framenet experts can efficiently label the sentences with frame structures. The most popular task regarding frame semantics is frame semantic parsing, a task of automatically extracting frame structures from sentences [39, 120]. Due to its popularity, a number of efforts have been made to enhance FrameNet by mapping it to other lexicons, such as WordNet, PropBank and VerbNet [50, 114, 152]. FrameNet+ [119] increased the lexical coverage of FrameNet through automatic paraphrasing and manual verification. Pancholy et al. [116] augmented FrameNet annotations by replacing annotated lexical units with unannotated lexical units in the same frame.

FrameNet has been used for a variety of tasks for knowledge learning. Baumgartner and Burchardt [9] transformed frames and frame relations into logic programs, potentially for applying inference on a larger scale. Aharon et al. [1] presented a template-based system to generate entailment rules. Yatskar et al. [191] introduced situation recognition, which is the problem of producing a concise summary of the situation that an image depicts. They use frames and their semantic roles as a representation to describe the main activity in the image.

Another popular lexical database in the field of NLP is WordNet [104]. It was first created at the Cognitive Science Laboratory of Princeton University in 1985, as a lexical database of semantic relations between words such as synonyms, hyponyms and meronymys. The creation of the database came from lexicographers' effort to cover all of the factual information about the meanings of each word. Some other commonsense knowledge bases import knowledge from WordNet into its own representation, such as ConceptNet [159].

2.1.2.2 Crowdsourcing

The construction of expert curated knowledge bases does not scale well due to its dependence on human experts. So attempts have been made to use crowd-sourcing techniques to compile knowledge bases of commonsense knowledge. The well-known project Open Mind Common Sense (OMCS), originally launched at the MIT Media Lab in 1999, collected commonsense knowledge from ordinary people through its website [157]. They designed a webpage that gathers facts, rules, stories, and descriptions using different interfaces and instructions. For example, one interface is to present the user with a simple story and ask for knowledge that is helpful in understanding the story, e.g., given the story "Bob had a cold. Bob went to the doctor.", the user might enter "Bob was feeling sick" or "Bob wanted to feel better". Thanks to the development of the world wide web, it has become possible for thousands of people to collaborate to construct systems that no single individual or team could build. OMCS has been running on the Web since September 2000 and they have gathered millions of pieces of commonsense knowledge from thousands of people.

ConceptNet [93] is a semantic network that originated from the Open Mind Common Sense project. It is represented as a knowledge graph that connects nodes (words and phrases of natural language) with labeled edges (relations). Since 2002, ConceptNet has experienced multiple updates. For example, it has since grown to include knowledge from other resources such as English Wiktionary and Wordnet [159]. ConceptNet 5.5, released in 2017, contains over 8 million nodes and 21 million edges, which are aligned with its pre-defined 36 relations such as AtLocation, Causes, and MadeOf. There are 2 relations closely related to our functional knowledge: the "Used For" relation which describes "A is used for B; the purpose of A is B" and the "Capable Of" relation that describes "Something that A can typically do is B." Chapter 4 will show detailed results using ConceptNet in our work.

ATOMIC [146] is a commonsense knowledge repository containing everyday commonsense inferential knowledge about events described in natural language represented by "if-then" relations. For example, if Person X compliments Person Y, then Person Y feels flattered; if Person X makes Person Y's coffee, then Person X gets thanked. They define 9 if-then relation types such as If-Event-Then-Mental-State and If-Event-Then-Event. ATOMIC consists of over 300k events associated with 877k inferential relations annotated by crowd workers. Their crowdsourcing framework gathers annotations in the form of free-form textual responses to simple questions in order to achieve large-scale and high-quality collection of commonsense about events.

2.1.2.3 Web Mining

Besides expert curation and crowd-sourcing, a number of previous works have attempted to learn commonsense knowledge from the Web in an unsupervised manner. Among them, knowledge graphs are probably the most widely studied topic. Knowledge graphs model information in the form of entities and relationships between them, where entities (e.g., people, object, location) are represented as nodes and relations (e.g., Is A, Be Located At) are represented as edges. One piece of knowledge is typically defined in the form of Subject-Predicate-Object (SPO) Triple, where subject and object are entities and predicate is the relation between them. For example, the triple (Oven, Be Located At, Kitchen) describes that the oven is typically in the kitchen. A large number of knowledge graphs have been created in the field, including Wikidata [175], Freebase [13], DBpedia [4], YAGO [161], and the Google Knowledge Graph [158].

One of the notable projects, the KnowItAll system [47] was developed at the University of Washington Turing Center headed by Oren Etzioni. The KnowItAll system aims to extract large collections of facts from the Web in an autonomous, unsupervised, and scalable manner following the paradigm of Open Information Extraction [48]. Different Open Information Extraction techniques have been applied to acquire different types of knowledge [91].

Another well-known system is NELL (Never-Ending Language Learner) [107], which was developed by the “Read the Web” project research team at Carnegie Mellon University. It has been designed to read the web 24 hours/day since January 2010 and have accumulated millions of candidate beliefs. It aims to extract knowledge of individual categories, e.g., how to map noun phrases into specific semantic categories, and the knowledge of the relation between different categories, e.g., beLocatedAt(object, location). It is worth noting that knowledge graphs do not necessarily focus on commonsense knowledge. However, methods that have been applied for building knowledge graphs have the potential for commonsense knowledge acquisition as well [93].

2.1.2.4 Language Models

Some previous publications tackled commonsense understanding using neural network models [143, 150]. More recently, due to the advancement in transfer learning [67, 139] and pre-trained large language models [43, 132], a considerable amount of work has attempted to learn commonsense knowledge from the pre-trained language models. Trinh and Le [171] applied large language models to mine commonsense facts from ConceptNet and their system outperforms previous state-of-the-art on the Winograd Schema Challenge. Davison et al. [41] used a pre-trained language model for commonsense knowledge base completion [70, 87]. They transform relational triples into masked sentences and then rank a triple’s validity by the estimated point-wise mutual information between the two entities generated by the masked language model. Petroni et al. [123] adopted a similar idea to conduct a systematic analysis of the commonsense knowledge in pretrained language models. COMET [16] is a transformer-based framework for automatic construction of commonsense knowledge bases. It was trained on ATOMIC and ConceptNet by adapting the weights of language models to learn to produce novel and diverse commonsense knowledge tuples. Bosselut et al. [17] used COMET to dynamically constructs a knowledge graph of commonsense knowledge on demand to provide context-dependent commonsense for downstream inference. More recently, ChatGPT [112] and GPT-4 [113] has attracted considerable attention within and outside the NLP community due to their strong ability in language understanding. More work needs to be done to understand their capability in commonsense understanding [88, 131].

2.2 Functional Knowledge

Functional knowledge is one specific type of commonsense knowledge that is essential for NLP tasks, such as narrative text understanding [83].

Pustejovsky [127] proposed *generative lexicon*, a theory of linguistic semantics that focuses on the distributed nature of compositionality in natural language. It defines a so called "Telic role" for entities referring to their purpose or function. Pustejovsky et al. [128] introduced the Brandeis Semantic Ontology, which was aimed as a large generative lexicon ontology and lexical database. More recently, Kazeminejad et al. [76] used rule-based methods to automatically extracts the telic roles from an ontology [111].

Another line of work closely related is the theory of *affordances*. The concept of perceptual affordances can be traced to the psychological research of Gibson, who proposed that organisms perceive the environment in terms of the action possibilities that they offer to them. Affordances reveal the functionalities of objects and the possible actions that can be performed on them. Gibson argued that when we look at a chair or a cup, our perception does not provide a generic perceptual view of these objects consisting of all of their qualities, but instead informs of the affordances such as sit-ability and lift-ability that they offer to us [75]. Sahin et al. [142] presented that each interaction of an agent with its environment can be represented as an affordance relation instance tuple: (entity, behavior, effect). McGregor and Jezek [101] discussed the importance of affordances in recognizing the meaning of the sentence from implicit verbs. For example, "patron enjoys beer" means "patron enjoys *drinking* beer" and "guest finishes cake" means "guest finishes *eating* cake". Chao et al. [29] collected affordances of objects through crowd-sourcing. Specifically, they asked the annotator whether it is possible (for a human) to perform the action on the object. An example annotation question is "Is it possible to *hunt* (pursue for food or sport (as of wild animals)) a *car*?". The worker needs to choose an answer from "Definitely yes", "Normally yes", "Maybe", "Normally no", "Definitely no", "I don't know", and "Description doesn't make sense or is grammatically incorrect".

Selectional preference (SP) could also provide a complementary view of the commonsense knowledge of object functions. Selectional Preference is a common phenomenon in human language that has been shown to be related to natural language understanding [182]. It refers to the phenomenon that, given a predicate (argument), people have preferences for which words are likely to be the argument (predicate) within a certain linguistic context. For example, *eat* prefers, as its object argument, words from the semantic class of food and disprefers words from the semantic class of fluids [89]. Zhang et al. [195] demonstrated that selectional preference is closely aligned with human commonsense knowledge. Specifically, they observed that the instances of "Used For" and "Capable Of" relations in ConceptNet can be well aligned with selectional preference triples manually annotated via crowd-sourcing. For example, a plausible SP pair (sing, song) can be mapped to (song, UsedFor, sing), (phone, ring) can be mapped to (phone, CapableOf, ring). This observation indicates that selectional preference may provide an effective way to acquire functional knowledge of objects. Modi et al. [108] observed that selectional preference can help predict upcoming content. Their experimental results showed that incorporating selectional preference can make the model predictions more similar to human expectations based on a cloze task. Another line of work in parallel with selectional preference is semantic plausibility [177]. It aims to distinguish a physically plausible event from an implausible one. For example, our commonsense knowledge helps us decide that a backpack can contain a cat, but not that a bottle can contain a cat.

2.3 Commonsense Knowledge of Objects

The knowledge of objects plays an essential part in our daily living and communication. Some work specifically argued the importance of commonsense knowledge about physical objects in natural language processing. For example, Hobbs et al. [66] introduced an axiomatization of what might be called "commonsense metaphysics", including time, space, physical objects, causality, functionality, etc. Physical objects are especially important for conceptual representation systems, such as conceptual dependency [147]. A number of following work of knowledge representation systems also showed the importance of the knowledge of physical objects [22, 83]. Lehnert and Burstein [83] presented a structure called Object Primitives, as an extension to the system of Conceptual Dependency for the purpose of representing physical objects and knowledge about the objects. They defined seven Object Primitives, such as *connector* (indicates what actions are normally enabled when an object is in a particular state) and *consumer* (describes objects whose primary function is to consume other objects).

The knowledge of physical object has been shown to benefit a plenty of different natural language understanding tasks, such as selectional preference [45], word sense disambiguation [2], language generation [106], virtual scene generation [28], and question answering [117]. The importance of this type of information has led to a considerable amount of attempts to learn different kinds of knowledge of physical objects.

Some previous publications attempted to acquire the knowledge of the characteristics of the objects. For example, Berland and Charniak [11] extracted parts of objects given a word denoting the whole object using a news corpus. They applied pre-defined patterns like "parts of wholes" and statistical method to extract parts of objects, like "engine of a car". Girju et al. [57] extends this work by allowing parts or wholes as compound nouns. They also introduce a more general method of iteratively formulating rules for a decision tree to determine the part-whole relations. Takamura and Tsujii [166] estimates numerical attributes of physical object, such as the size, length, and height. They utilize both absolute clues from a search engine, as well as relative clues such as "A is larger than B" from corpus.

There has also been considerable amount of work trying to learn different relations between objects.

Yatskar et al. [190] proposed a model to extract visual commonsense facts of objects from annotated images and their annotated descriptions using point-wise mutual information. Their work focused on spatial information of co-occurring objects through six types of unique relations (touches, above, besides, holds, on, disconnected). Forbes and Choi [54] present an approach to learn relative physical knowledge of actions and objects along five dimensions (e.g., size, weight, strength, rigidness, and speed) from unstructured natural language text. Gao et al. [55] tried to learn the relations between concrete actions in the form of verb and noun pairs, as well as their effects on the state change of physical objects. Their action-effect prediction was modeling in a multimodal fashion.

Collell et al [35] introduced the task of predicting relative spatial arrangements of two objects under a relationship, which requires common sense spatial knowledge about objects and actions. Their spatial templates are not restricted to explicit prepositions such as on, above, below, but also allow implicit spacial relations such as "man pulling luggage". More generally, SpRL and SpaceEval [77, 129] introduced a task of identifying and categorizing spacial information, including not only static spatial relations, but also dynamic relations such as motion indicators. More recently, Manotas et al. [98] introduced a larger dataset specifically for identifying motion of physical entities in text, which contains 1,200 motion verbs, compared to the four motion verbs (e.g., arrive, leave, drive, and walk) in SpaceEval.

Xu et al. [187] presented the "LocatedNear" relation as one type of commonsense knowledge describing two physical objects that are typically found near each other. For example, table and chair, door and bell, window and building. They propose LSTM-based neural models to automatically extract the knowledge from the Project Gutenberg corpus [80], which contains 3,036 English books by 142 authors. They released a dataset containing 500 pairs of objects with human annotation for how likely the objects are to appear near each other.

CHAPTER 3

ACQUIRING PROTOTYPICAL FUNCTIONS FOR LOCATIONS

Every day, people go to different places to accomplish goals. People go to stores to buy clothing, go to restaurants to eat, and go to the doctor for medical services. People travel to specific destinations to enjoy the beach, go skiing, or see historical sites. For most places, people typically go there for a common set of reasons, which we will refer to as **functions** of the location.[1] Specifically, we call them *prototypical goal activities (goal-acts)* for a location. For example, a prototypical goal-act for restaurants would be "*eat food*" and for IKEA would be "*buy furniture*." In this chapter, we will introduce a technique to automatically acquire the knowledge of prototypical goal activities.

Previous research has established that recognizing people's goals is essential for narrative text understanding and story comprehension [46, 60, 82, 149, 181]. Goals and plans are essential to understand people's behavior, and we use our knowledge of prototypical goals to make inferences when reading. For example, consider the following pair of sentences: *"Mary went to the supermarket. She needed milk."* Most people will infer that Mary purchased milk, unless told otherwise. But a purchase event is not explicitly mentioned. In contrast, a similar sentence pair *"Mary went to the theatre. She needed milk."* feels incongruent and does not produce that inference. Recognizing goals is also critical for conversational dialogue systems. For example, if a friend tells you that they went to a restaurant, you might reply *"What did you eat?"*, but if a friend says that they went to Yosemite, a more appropriate response might be *"Did you hike?"* or *"Did you see the waterfalls?"*.

Our knowledge of prototypical goal activities also helps us resolve semantic ambiguity. For example, consider the following sentences:

[1] See more discussions on the scope of a "location" in Section 3.1.1.

```
(a) She went to the kitchen and got chicken.
(b) She went to the supermarket and got chicken.
(c) She went to the restaurant and got chicken.
```

In sentence (a), we infer that she retrieved chicken (e.g., from the refrigerator) but did not pay for it. In (b), we infer that she paid for the chicken but probably did not eat it at the supermarket. In (c), we infer that she ate the chicken at the restaurant. Note how the verb "*got*" maps to different presumed events depending on the location.

Our research aims to learn the prototypical goal-acts for locations using a text corpus. We propose a learning framework that consists of three steps. First, we extract activities that co-occur with locations in goal-oriented syntactic patterns. Next, we construct an *activity profile matrix* that consists of an activity vector (profile) for each of the locations. We then apply a semi-supervised label propagation algorithm to iteratively revise the activity profile strengths based on a small set of labeled locations. We also incorporate external resources to measure similarity between different activity expressions. Our results show that this semi-supervised learning approach outperforms several baseline methods in identifying the prototypical goal activities for locations.

3.1 Learning Prototypical Goal Activities

Our aim is to learn the most prototypical goal-acts for locations. To tackle this problem, we first extract locations and related activities from a large text corpus. Then we use a semi-supervised learning method to identify the goal activities for individual locations. In the following sections we describe these processes in detail.

3.1.1 Location and Activity Extraction

To collect information about locations and activities, we use the 2011 Spinn3r dataset [23]. Since our interest is learning about the activities of ordinary people in their daily lives, we use the Weblog subset of the Spinn3r corpus, which contains over 133 million blog posts.

We use the text data to identify activities that are potential goal-acts for a location. However we also need to identify locations and want to include both proper names (e.g., Disneyland) as well as nominals (e.g., store, beach), so Named Entity Recognition will not suffice. Consequently, we extract (Loc, Act) pairs using syntactic patterns.

First, we apply the Stanford dependency parser [97]. We extract the lemmatized sentences that match the pattern "go to X to Y" with the following conditions:

(1) there exists a subject connecting to "go",

(2) X has an **nmod** (nominal modifier) relation to "go" (lemma),

(3) X is a noun or noun compound,

(4) Y has an **xcomp** relation (open clausal complement) with "go",

(5) Y is a verb.

Figure 3.1 depicts the intended syntactic structure, which we will informally call the "go to" pattern. For sentences that match this pattern, we extract X as a location and Y as an activity. If the verb is followed by a particle and/or noun phrase (NP), then we also include the particle and head noun of the NP. For example, we extract activities such as *"pray"*, *"clean up"*, and *"buy sweater"*.

This syntactic structure was chosen to identify activities that are described as being the reason why someone went to the location. However it is not perfect. In some cases, X is not a location (e.g., *"go to great lengths to ..."* yields "*lengths*" as a location), or Y is not a goal-act for X (e.g., *"go to the office to retrieve my briefcase ..."* yields "*retrieve briefcase*" which is not a prototypical goal for "*office*"). Interestingly, the pattern extracts some nominals that are not locations in a strict sense, but behave as locations. For example, *"go to the doctor"* extracts "*doctor*" as a location. Literally a doctor is a person, but in this context it really refers to the doctor's office, which is a location. The pattern also extracts entities such as "*roof*", which are not generally thought of as locations but do have a fixed physical location. Other extracted entities are virtual but function as locations, such as "*Internet*". For the purposes of this work, we use the term **location** in a general sense to include any

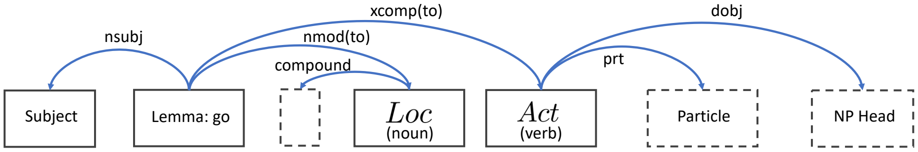


Figure 3.1: Dependency relation structure for "go to" pattern.

place or object that has a physical, virtual or implied location.

The "go to" pattern worked quite well at extracting (Loc, Act) pairs, but in relatively small quantities due to the very specific nature of the syntactic structure. So we tried to find additional activities for those locations. Initially, we tried harvesting activities that occurred in close proximity (within 5 words) to a known location, but the results were too noisy. Instead, we used the pattern "Y in/at X" with the same syntactic constraints for Y (the extracted activity) and X (a location extracted by the "go to" pattern).

We discovered many sentences in the corpus that were exactly or nearly the same, differing only by a few words, which resulted in artificially high frequency counts for some (Loc, Act) pairs. So we filtered duplicate or near-duplicate sentences by computing the longest common substring of sentence pairs that extracted the same (Loc, Act). If the shared substring had length ≥ 5, then we discarded the "duplicate" sentence.

Finally, we applied three filters. To keep the size of the data manageable, we discarded locations and activities that were each extracted with frequency < 30 by our patterns, i.e., remove too rare locations with $|(Loc, *)| < 30$ and too rare activities with $|(*, Act)| < 30$. And we manually filtered locations that are Named Entities corresponding to cities or larger geo-political regions (e.g., provinces or countries). Large regions defined by government boundaries fall outside the scope of our task because the set of activities that typically occur in (say) a city or country is so broad. Finally, we added a filter to try to remove extremely general activities that can occur almost anywhere (e.g., visit). If an activity co-occurred with $> 20\%$ of the extracted (distinct) locations, then we discarded it.

After these filters, we extracted 451 distinct locations, $5,143$ distinct activities, roughly $200,000$ distinct (Loc, Act) pairs, and roughly $500,000$ instances of (Loc, Act) pairs as shown in Table 3.1.

3.1.2 Activity Profiles for Locations

We define an *activity profile matrix* Y of size $n \times m$, where n is the number of distinct locations and m is the number of distinct activities. The rows correspond to different locations and columns represent different activities. Note that the order of activities are always the same across different rows, e.g., the second activity in row #1 is the same as the second activity in row #100. $Y_{i,j}$ represents the strength of the jth activity a_j being a

Table 3.1: Statistics of the dataset.

	Count
# of distinct locations	451
# of distinct activities	5,143
# of distinct location and activity pairs	180,957
# of all location and activity pairs	524,352

goal-act for l_i. For example, as illustrated in Figure 3.2, $Y_{i,j}$ is represented by a circle with different colors. Location *cinema* is strongly related to *watch film* (as indicated by red) and moderately related to *eat popcorn* (as indicated by light red). *Theatre* sometimes refer to the movie theatre so that *watch film* is the goal activity (in red), but it can also refer to musical theatre or broadway theatre when the main activity should be *watch show* (also in red). While *Sea World* is strongly related to *watch show*, not likely *watch film* (in grey). We use $\mathbf{y}_i \in \mathbb{R}^m$ to denote the ith row of Y.

Table 3.2 shows a more concrete example of (partial) activity profiles for 4 locations and 3 activities with strength values.[2] Our goal is to learn the $Y_{i,j}$ values so that activities with high strength are truly goal-acts for location l_i.

We could build the activity profile for location l_i using the co-occurrence data extracted from the blog corpus. For example, we could estimate $P(a_j \mid l_i)$ directly from the frequency

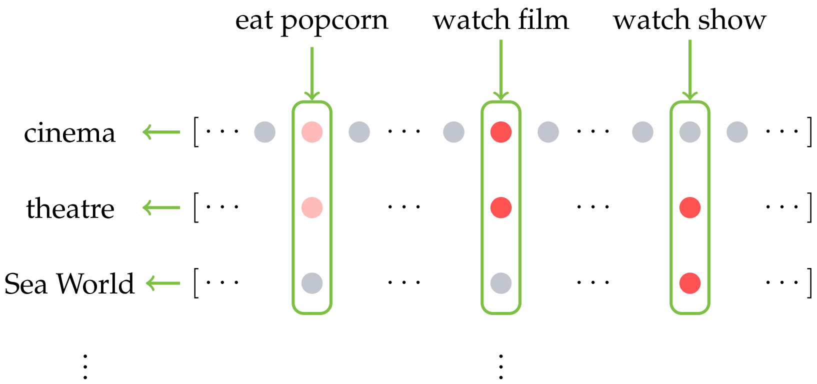


Figure 3.2: Each location (row) has an *activity profile*, where each column corresponds to one activity. Each circle represents a value (between 0 and 1) indicating the strength of the column activity being a goal activity for row location. Color red represents a strong relation.

[2] Not actual values, for illustration only.

Table 3.2: An illustration of the activity profile matrix Y.

	$a_1 =$ buy book	$a_2 =$ eat burger	...	$a_m =$ pray
$l_1 =$ McDonald's	.10	**.30**		.01
$l_2 =$ Burger King	.12	**.50**		.02
$l_3 =$ bookstore	**.40**	.02		.04
$\vdots$		$\vdots$		
$l_n =$ church	.05	.01		**.70**

counts of the activities extracted for l_i. However, a high co-occurrence frequency doesn't necessarily mean that the activity represents a prototypical goal. For example, the activity *"have appointment"* frequently co-occurs with *"clinic"* but doesn't reveal the underlying reason for going to the clinic (e.g., probably to see a doctor or undergo a medical test). To appreciate the distinction, imagine that you asked a friend why she went to a health clinic, and she responded with "because I had an appointment". You would likely view her response as being snarky or evasive (i.e., she didn't want to tell you the reason). In Section 3.2, we will evaluate this approach as a baseline and show that it does not perform well.

3.1.3 Semi-Supervised Learning of Goal-Act Probabilities

Our aim is to learn the activity profiles for locations using a small amount of labeled data, so we frame this problem as a semi-supervised learning task. Given a small number of "seed" locations coupled with predefined goal-acts, we want to learn the goal-acts for new locations.

3.1.3.1 Location Similarity Graph

We use $l_i \in L$ to represent location l_i, where $|L| = n$. We define an undirected graph $G = (V, E)$ with vertices representing locations ($|V| = n$) and edges $E = V \times V$, such that each pair of vertices v_i and v_k is connected with an edge e_{ik} whose weight represents the similarity between l_i and l_k.

We can then represent the edge weights as an $n \times n$ symmetric weight matrix W indicating the similarity between locations. There could be many ways to define the weights, but for now we use the following definition from [199], where σ^2 is a hyper-parameter[3]:

[3] We use the same value $\sigma^2 = 0.03$ as [199].

$$W_{i,k} = \exp\left(-\frac{1}{\sigma^2}\left(1 - sim\left(l_i, l_k\right)\right)\right) \tag{3.1}$$

To assess the similarity between locations, we measure the cosine similarity between vectors of their co-occurrence frequencies with activities. Specifically, let matrix $F_{n \times m} = [\mathbf{f}_1, ..., \mathbf{f}_n]^{\mathrm{T}}$ where $\mathbf{f}_i$ is a vector of length m capturing the co-occurrence frequencies between location l_i and each activity a_j in the extracted data (i.e., $F_{i,j}$ is the number of times that activity a_j occurred with location l_i). We then define location similarity as:

$$sim(l_i, l_k) = \frac{\mathbf{f}_i^{\mathrm{T}} \mathbf{f}_k}{\|\mathbf{f}_i\| \|\mathbf{f}_k\|} \tag{3.2}$$

3.1.3.2 Initializing Activity Profiles

We use semi-supervised learning with a set of “seed” locations from human annotations, and another set of locations that are unlabeled. So we subdivide the set of locations into $S = \{l_1, ..., l_s\}$, which are the seed locations, and $U = \{l_{s+1}, ..., l_{s+u}\}$, which are the unlabeled locations, such that $s + u = n$. For an unlabeled location $l_i \in U$, the initial activity profile is the normalized co-occurrence frequency vector $\bar{\mathbf{f}}_i$.

For each seed location $l_i \in S$, we first automatically construct an activity profile vector $\bar{\mathbf{h}}_i$ based on the gold goal-acts which were obtained from human annotators as described in Section 3.2.1. All activities not in the gold set are assigned a value of zero. Each activity a_j in the gold set is assigned a probability $P(a_j \mid l_i)$ based on the gold answers. However, the gold goal-acts may not match the activity phrases found in the corpus (see discussion in Section 3.2.3), so we smooth the vector created with the gold goal-acts by averaging it with the normalized co-occurrence frequency vector $\bar{\mathbf{f}}_i$ extracted from the corpus.

The activity profiles of seed locations stay constant through the learning process. We use $\mathbf{y}_i^0$ to denote the initial activity profiles. So when $l_i \in S$, $\mathbf{y}_i^0 = (\bar{\mathbf{f}}_i + \bar{\mathbf{h}}_i)/2$.

3.1.3.3 Learning Goal-Act Strengths

We apply a learning framework developed by [199] based on harmonic energy minimization and extend it to multiple labels. Intuitively, we assume that similar locations should share similar activity profiles,[4] which motivates the following objective function

[4] This is a heuristic but is not always true.

over matrix Y:

$$\begin{aligned} &\arg\min_{Y} \sum_{i,k} W_{i,k} \|\mathbf{y}_i - \mathbf{y}_k\|^2, \\ &\text{s.t. } \mathbf{y}_i = \mathbf{y}_i^0 \text{ for each } l_i \in S \end{aligned} \tag{3.3}$$

Let $D = (d_i)$ denote an $n \times n$ diagonal matrix where $d_i = \sum_{k=1}^{n} W_{i,k}$. Let's split Y by the sth row: $Y = \begin{bmatrix} Y_s \\ Y_u \end{bmatrix}$, then split W(similarly for D) into four blocks by the sth row and column:

$$W = \begin{bmatrix} W_{ss} & W_{su} \\ W_{us} & W_{uu} \end{bmatrix} \tag{3.4}$$

From [199], Eq (3.3) is given by:

$$Y_u = (D_{uu} - W_{uu})^{-1} W_{us} Y_s \tag{3.5}$$

We then use the label propagation algorithm described in [198] to compute Y:

```
Algorithm 1
  repeat
    $Y \leftarrow D^{-1}WY$
    Clamp $\mathbf{y}_i = \mathbf{y}_i^0$ for each $l_i \in S$
  until convergence
```

3.1.3.4 Activity Similarity

One problem with the above algorithm is that it only takes advantage of relations between vertices (i.e., locations). If there are intrinsic relations between activities, they could be exploited as a complementary source of information to benefit the learning. Intuitively, different pairs of activities share different similarities, e.g., "*eat burgers*" should be more similar to "*have lunch*" than "*read books*."

Under this idea, similar to the previous location similarity weight matrix W, we want to define an activity similarity weight matrix $A_{m \times m}$ where $A_{i,k}$ indicates the similarity weight between activity a_i and a_k:

$$A_{i,k} = \exp\left(-\frac{1}{\sigma^2}\left(1 - sim\left(a_i, a_k\right)\right)\right) \tag{3.6}$$

where σ^2 is the same as in Eq (3.1).

We explore 3 different similarity functions $sim(a_i, a_k)$ based on co-occurrence with locations, word matching, and embedding similarities.

First, similar to Eq (3.2), we can use each activity's co-occurrence frequency with all locations as its location profile and define a similarity score based on cosine values of location profile vectors:

$$sim^L(a_i, a_k) = \frac{\mathbf{g}_i^{\mathrm{T}} \mathbf{g}_k}{\|\mathbf{g}_i\| \|\mathbf{g}_k\|} \tag{3.7}$$

where the predefined co-occurrence frequency matrix $F = [\mathbf{f}_1, ..., \mathbf{f}_n]^{\mathrm{T}} = [\mathbf{g}_1, ..., \mathbf{g}_m]$.

As a second option, the similarity between activities can often be implied by their lexical overlap, e.g., two activities sharing the same verb or noun might be related. For each word belonging to any of our activities, we use WordNet [104] to find its synonyms. We also include the word itself in the synonym set. If the synonym sets of two words overlap, we call these two words "**match**". Then we define the lexical overlap similarity function between a_i and a_k:

$$sim^O(a_i, a_k) = \begin{cases} 1 & \text{if verb and noun match} \\ 0.5 & \text{if verb or noun match} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

As a third option, we can use 300-dimension word embedding vectors [121] trained on 840 billion tokens of web data to compute semantic similarity. We compute an activity's embedding as the average of its words' embeddings. Let $sim^E(a_i, a_k)$ be the cosine value between the embedding vectors of a_i and a_k:

$$sim^E(a_i, a_k) = cos\langle \text{Embed}(a_i), \text{Embed}(a_k) \rangle \tag{3.9}$$

Finally, we can plug these similarity functions into Eq (3.6). We use A^L, A^O, A^E to denote the corresponding matrix. We can also plug in multiple similarity metrics such as $(sim^L + sim^E)/2$ and use combination symbols A^{L+E} to denote the matrix.

3.1.3.5 Injecting Activity Similarity

Once we have a similarity matrix for activities, the next question is how will it help with the activity profile computation? Recall from Eq (3.5), we know that the activity profile of an unlabeled location can be represented by a linear combination of other locations' activity profiles. The activity profile matrix Y is an $n \times m$ matrix where each row is the activity profile for a location. We can also view Y as a matrix whose each column is the location profile for an activity. Using the same idea, we can make each column approximate a linear combination of its highly related columns (i.e., the location profile of an activity will become more similar to the location profiles of its similar activities). Our expectation is that this approximation will help improve the quality of Y.

By being right multiplied by matrix A, Y gets updated from manipulating its columns (activities) as well. We modify the algorithm accordingly as below:

Algorithm 2

```
repeat
    $Y \leftarrow D^{-1}WYA$
    Clamp $\mathbf{y}_i = \mathbf{y}_i^0$ for each $l_i \in S$
until convergence
```

3.2 Evaluation

3.2.1 Gold Standard Data

Since this is a new task and there is no existing dataset for evaluation, we use crowdsourcing via Amazon Mechanical Turk (AMT) to acquire gold standard data. First, we released a qualification test containing 15 locations along with detailed annotation guidelines. 25 AMT workers finished our assignment, and we chose 15 of them who did the best job following our guidelines to continue. We gave the 15 qualified workers 200 new locations, consisting of 152 nominals and 48 proper names,[5] randomly selected from our extracted data and set aside as test data. For each location, we asked the AMT workers to complete the following sentence:[6]

[5] Same distribution as in the whole location set.

[6] The annotation guidelines can be found in Appendix B.

People go to *LOC* to ______ ______

VERB NOUN

LOC was replaced by one of the 200 locations. Annotators were asked to provide an activity that is the primary reason why a person would go to that location, in the form of just a VERB or a VERB NOUN pair. Annotators also had the option to label a location as an "ERROR" if they felt that the provided term is not a location, since our location extraction was not perfect.

Only 10 annotators finished labeling our test cases, so we used their answers as the gold standard. We discarded 12 locations that were labeled as an "ERROR" by ≥ 3 workers.[7] This resulted in a test set of 188 locations paired with 10 manually defined goal-acts for each one.

A key question that we wanted to investigate through this manual annotation effort is to know whether people truly do associate the same prototypical goal activities with locations. To what extent do people agree when asked to list goal-acts? Also, some places clearly have a smaller set of goal-acts than others. For example, the primary reason to go to an airport is to catch a flight, but there's a larger set of common reasons why people go to Yosemite (e.g.,"*hiking camping*", "*rock climbing*", "*see waterfalls*", etc.). So we examined all 200 locations and check how often do annotators agree with each other in Figure 3.3. The x-axis represents the maximum number of annotators that provide the same answer for one location. The y-axis represents the percentage of locations. For example, the $x = 2$ column means that about 90% of locations have 2 out of 10 annotators providing the same answer and there are no 3 answers that are the same.

Complicating matters, the AMT workers often described the same activity with different words (e.g., "*buy book*" vs. "*purchase book*"). Automatically recognizing synonymous event phrases is a difficult NLP problem in its own right.[8] So solely for the purpose of analysis, we manually merged activities that have a nearly identical meaning.[9] We

[7] We found that the workers rarely used the "ERROR" label, so setting this threshold to be 3 was a strong signal.

[8] We tried using WordNet synsets to conflate phrases, but it didn't help much.

[9] The list of merged activities can be found in Appendix A.

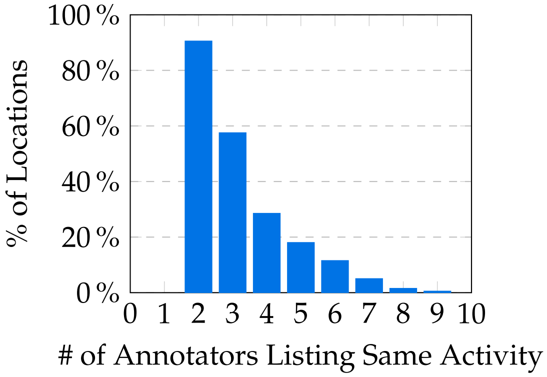


Figure 3.3: x-axis is the maximum number of the same activity provided by 10 annotators for one location, and y-axis is the percent of locations whose maximum number of the same activity $\geq x$.

were extremely conservative and did not merge similar or related phrases that were not synonymous because the granularity of terms may matter for this task (e.g., we did not merge "*eat burger*" and "*eat lunch*" because one may apply to a specific location while the other does not).

Figure 3.4 shows the results of our analysis. Only 1 location was assigned exactly the same goal-act by all 10 annotators. But at least half (5) of the annotators listed the same goal-act for 40% of the locations. And nearly 80% of locations had one or more goal-acts listed by $\geq$ 3 people. These results show that people often do share the same associations between prototypical goal-acts and locations. These results are also very conservative because many different answers were also similar (e.g. "*eat burger*", "*eat meal*").

In Table 3.3, we show examples of locations and the goal-acts listed for them by the

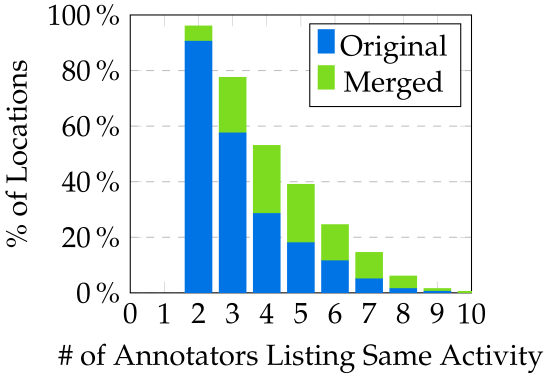


Figure 3.4: Percentage of locations that have at least one goal-act assigned by multiple annotators. The stacked red bars represent the location percentage after merging activities.

Table 3.3: Goal-acts provided by human annotators.

Location	Gold Goal-Acts
Toys R Us	buy toys (9), browse gifts
sink	wash hands (7), wash dishes (3)
airport	catch flight (7), board planes, take airplane, take trips
bookstore	buy books (6), browse books (2), browse best-sellers, read book
lake	go fishing (3), go swimming (3), drive boat (2), ride boat, see scenery
chiropractor	get treatment (3), adjust backs (3), alleviate pain (2), get adjustment, get aligned
Chinatown	buy goods (2), buy duck, buy souvenirs, eat dim sum, eat rice, eat wontons, find Chinese, speak Chinese, visit restaurants

human annotators. If multiple people gave the same answer, we show the number in parentheses. For example, given the location "Toys R Us", 9 people listed "*buy toys*" as a goal-act and 1 person listed "*browse gifts*". We see from Table 3.3 that some locations yield very similar sets of goal-acts (e.g., sink, airport, bookstore), while other locations show more diversity (e.g., lake, chiropractor, Chinatown).

3.2.2 Baselines

To assess the difficulty of this NLP task, we created 3 baseline systems for comparison with our learning approach. All of these methods take the list of activities that co-occurred with a location l_i in our extracted data and rank them.

The first baseline, **FREQ**, ranks the activities based on the co-occurrence frequency $F_{i,j}$ between l_i and a_j in our patterns. The second baseline, **PMI**, ranks the activities using point-wise mutual information. The third baseline, **EMBED**, ranks the activities based on the cosine similarity of the semantic embedding vectors for l_i and a_j. We use GloVe [121] 300-dimension embedding vectors pre-trained on 840 billion tokens of web data. For locations and activities with multiple words, we create an embedding by averaging the vectors of their constituent words.

3.2.3 Matching Activities

The gold standard contains a set of goal-acts for each location. Since the same activity can be expressed with many different phrases, the only way to truly know whether two phrases refer to the same activity is manual evaluation, which is expensive. Furthermore, many activities are very similar or highly related, but not exactly the same. For example, "*eat burger*" and "*eat food*" both describe eating activities, but the latter is more general than the former. Considering them to be the same is not always warranted (e.g., "*eat burger*" is a logical goal-act for McDonald's but not for Baskin-Robbins which primarily sells ice cream). As another example, "*buy chicken*" and "*eat chicken*" refer to different events (buying and eating) so they are clearly not the same semantically. But at a place like KFC, buying chicken implies eating chicken, and vice versa, so they seem like equally good answers as goal-acts for KFC. Due to the complexities of determining which gold standard answers belong in equivalence classes, we considered all of the goal-acts provided by the human annotators to be acceptable answers.

To determine whether an activity a_j produced by our system matches any of the gold goal-acts for a location l_i, we report results using two types of matching criteria. **Exact Match** judges a_j to be a correct answer for l_i if (1) it exactly matches (after lemmatization) any activity in l_i's gold set, or (2) a_j's *verb* and *noun* both appear in l_i's gold set, though possibly in different phrases. For example, if a gold set contains "*buy novels*" and "*browse books*", then "*buy books*" will be a match.

Since Exact Match is very conservative, we also define a **Partial Match** criterion to give 50% credit for answers that partially overlap with a gold answer. An activity a_j is a partial match for l_i if either its *verb* or *noun* matches any of the activities in l_i's gold set of goal-acts. For example, "*buy burger*" would be a partial match with "*buy food*" because their verbs match.

3.2.4 Evaluation Metrics

All of our methods produce a ranked list of hypothesized goal-acts for a location. So we use Mean Reciprocal Rank (MRR) to judge the quality of the top 10 activities in each ranked list. We report two types of MRR scores.

MRR based on the Exact Match criteria (MRR_E) is computed as follows, where n is the number of locations in the test set:

$$\mathrm{MRR_E} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\text{rank of } 1^{st} \text{ Exact Match}} \tag{3.10}$$

We also compute MRR using both the Exact Match and Partial Match criteria. First, we need to identify the "best" answer among the 10 activities in the ranked list, which depends both on each activity's ranking and its matching score. The matching score for activity a_j is defined as:

$$\mathrm{score}(a_j) = \begin{cases} 1 & \text{if } a_j \text{ is an Exact Match} \\ 0.5 & \text{if } a_j \text{ is a Partial Match} \\ 0 & \text{otherwise} \end{cases}$$

Given 10 ranked activities a_1 ... a_{10} for l_i, we then compute:

$$\text{best\_score}(l_i) = \max_{j=1..10}\frac{\mathrm{score}(a_j)}{\mathrm{rank}(a_j)}$$

And then finally define $\mathrm{MRR_P}$ as follows:

$$\mathrm{MRR_P} = \frac{1}{n}\sum_{i=1}^{n}\text{best\_score}(l_i) \tag{3.11}$$

In order to get a more intuitive understanding of the model performance, we also compute a hard **Overlap** rate – the percentage of locations whose top 10 activities contains at least one Exact Match.

3.2.5 Experimental Results

Unless otherwise noted, all of our experiments report results using 4-fold cross-validation on the 200 locations in our test set. We used 4 folds to ensure 50 seed locations for each run (i.e., 1 fold for training and 3 folds for testing).

The first two columns of Table 3.4 show the MRR results under Exact Match and Partial Match conditions. The first 3 rows show the results for the baseline systems, and the remaining rows show results for our Activity Profile (AP) semi-supervised learning method. We show results for 5 variations of the algorithm: **AP** uses Algorithm 1, and the others use Algorithm 2 with different Activity Similarity measures: **AP+A$^{\mathbf{L}}$** (location profile similarity), **AP+A$^{\mathbf{O}}$** (overlap similarity), **AP+A$^{\mathbf{E}}$** (embedding similarity), and **AP+A$^{\mathbf{L+E}}$** (location

Table 3.4: Scores for MRR, Overlap and Top *k* results.

	MRR_E	MRR_P	Overlap	TOP1	TOP2	TOP3
EMBED	0.02	0.09	0.39	0.05	0.08	0.12
PMI	0.20	0.33	0.77	0.25	0.36	0.41
FREQ	0.23	0.34	0.76	0.23	0.32	0.40
AP	0.28	0.38	0.83	0.29	0.41	0.47
AP+A^L	0.28	0.40	0.82	0.32	**0.44**	0.49
AP+A^O	0.23	0.33	0.80	0.24	0.35	0.43
AP+A^E	0.25	0.36	0.85	0.28	0.40	0.47
AP+A^{L+E}	**0.29**	**0.42**	**0.85**	**0.35**	**0.44**	**0.52**

profiles plus embeddings).

Table 3.4 shows that our AP algorithm outperforms all 3 baseline methods. When adding Activity Similarity into the algorithm, we find that A^L slightly improves performance, but A^O and A^E do not. However, we also tried combining them and obtained improved results by using A^L and A^E together, yielding an MRR_P score of 0.42.

To gain more insight about the behavior of the models, Table 3.4 also shows results for the top-ranked 1, 2, and 3 answers. For these experiments, the system gets full credit if any of its top *k* answers exactly matches the gold standard, or 50% credit if a partial match is among its top *k* answers. These results show that our AP method produces more correct answers at the top of the list than the baseline methods.

Table 3.5 shows six locations with their gold answers and the Top 3 goal-acts hypothesized by our best AP system and the PMI and FREQ baselines. The activities in **boldface** were deemed correct (including Partial Match). For "bookstore" and "pharmacy", all of the methods perform well. Note the challenge of recognizing that different phrases mean essentially the same thing (e.g., "*fill prescription*", "*pick up prescription*", "*find medicine*"). For "university" and "Meijer", the AP method produces more appropriate answers than the baseline methods. For "market" and "phone", all three methods struggle to produce good answers. Since "market" is polysemous, we see activities related to both stores and financial markets. And "phone" arguably is not a location at all, but most human annotators treated it as a virtual location, listing goal-acts related to telephones. However our algorithm considered phones to be similar to computers, which makes sense for today's smartphones. In general, we also observed that Internet sites behave as virtual locations

Table 3.5: Examples of Top 3 hypothesized prototypical goal activities.

Location	**Gold Activity List**	**AP+A^{L+E} Top 3**	**PMI Top 3**	**FREQ Top 3**
bookstore	buy book (6) browse book (2) browse bestseller read book	**buy book** **purchase book** **see book**	**buy copy** **purchase book** **buy book**	**buy book** **browse** **find book**
pharmacy	get drug (4) fill prescription (3) get prescription (2) buy medicine	**find medicine** **get prescription** **pick up prescription**	**buy pill** **fill prescription** **pick up prescription**	**buy pill** **fill prescription** **pick up prescription**
university	get degree (4) gain education (5) watch sport	**gain education** **further education** **gain knowledge**	study law study psychology pursue study	enrol[10] enroll take class
Meijer	buy grocery (8) buy cream obtain grocery	**buy item** go shopping get item	check out deal have shopping post today	get item save money check out
market	buy grocery (6) buy fresh, buy goods buy shirt, find produce	make money eat out eat lunch	have demand increase competition lead player	trade intervene make money
phone	make call (4), ERROR (2) answer call, call friend have conversation stop ring	play game browse website view website	put number have number put card	plug glance have number

in language (e.g., "*I went to YouTube...*").

3.2.6 Discussion

The goal-acts learned by our system were extracted from the Spinn3r dataset, while the gold standard answers were provided by human annotators, so the same (or very similar) activities are often expressed in different ways (see Section 3.2.3). This raises the question: what is the upper bound on system performance when evaluating against human-provided goal-acts? To answer this, we compared all of the activities that co-occurred with each location in the corpus against its gold goal-acts. Only 36% of locations had at least one gold goal-act among its extracted activities when matching identical strings (after lemmatization). Because of this issue, our Exact Match criteria also allowed for combining verbs and nouns from different gold answers. Under this Exact Match criteria, 73% of locations had at least one gold goal-act among the extracted activities, so this represents an upper bound on performance using this metric. Under the Partial Match criteria, 98% of locations had at least one gold goal-act among the extracted activities, but only 50% credit was awarded for these cases so the maximum score possible would be ∼86%.

[10] A lemmatization error for the verb "enrolled".

We also manually inspected 200 gold locations to analyze their types. We discovered some related groups, but substantial diversity overall. The largest group contains $\sim$20% of the locations, which are many kinds of stores (e.g., Ikea, WalMart, Apple store, shoe store). Even within a group, different locations often have quite different sets of co-occurring activities. In fact, we discovered some spelling variants (e.g., "WalMart" and "wal mart"), but they also have substantially different activity vectors (e.g., because one spelling is much more frequent), so the model learns about them independently.[11] Other groups include restaurants ($\sim$5%), home-related (e.g., bathroom) ($\sim$5%), education ($\sim$5%), virtual (e.g., Wikipedia) ($\sim$3%), medical ($\sim$3%) and landscape (e.g., hill) ($\sim$3%). It is worth noting that our locations were extracted by two syntactic patterns and it remains to be seen if this has brought in any bias — detecting location nouns (especially nominals) is a challenging problem in its own right.

3.3 Related Work

Recognizing plans and goals is fundamental to narrative story understanding [19, 149]. Conceptual knowledge structures developed in prior work have shown the importance of this type of knowledge, including plans [181], goal trees [25], and plot units [82]. Wilensky's research aimed to understand the actions of characters in stories by analyzing their goals, and their plans to accomplish those goals. For example, someone's goal might be to obtain food with a plan to go to a restaurant. Our work aims to learn prototypical goals associated with a location, to support similar inference capabilities during story understanding.

Goals and plans can also function to trigger *scripts* [37], such as the $RESTAURANT script. There has been growing interest in learning narrative event chains and script knowledge from large text corpora [26, 27, 69, 124, 125]. In addition, Goyal et al. [59, 60] developed a system to automatically produce plot unit representations for short stories. A manual analysis of their stories revealed that 61% of Positive/Negative Affect States originated from completed plans and goals, and 46% of Mental Affect States originated from explicitly stated or inferred plans and goals.

Elson and McKeown [46] included plans and goals in their work on creating exten-

[11] Of course normalizing location names beforehand may be beneficial in future work.

sive story bank annotations that capture the knowledge needed to understand narrative structure. Researchers have also begun to explore NLP methods for recognizing the goals, desires, and plans of characters in stories. Recent work has explored techniques to detect wishes (desires) in natural language text [58] and identify desire fulfillment [31, 135].

Graph-based semi-supervised learning has been successfully used for many tasks, including sentiment analysis [49, 137], affective event recognition [44] and class-instance extraction [169]. The semi-supervised learning algorithm used in this chapter is modeled after a framework developed by Zhu et al. [199] based on harmonic energy minimization and a label propagation algorithm described in [198].

3.4 Conclusion

We introduced the problem of learning prototypical goal activities for locations. We obtained human annotations and showed that people do associate prototypical goal-acts with locations. We then created an activity profile framework and applied a semi-supervised label propagation algorithm to iteratively update the activity strengths for locations. We demonstrated that our learning algorithm identifies goal-acts for locations more accurately than several baseline methods.

However, this problem is far from solved. Challenges also remain in how to evaluate the accuracy of goal knowledge extracted from text corpora. Nevertheless, our work represents a first step toward learning goal knowledge about locations, and we believe that learning knowledge about plans and goals is an important direction for natural language understanding research. In future work, we hope to see if we can take advantage of more contextual information as well as other external knowledge to improve the recognition of goal-acts.

CHAPTER 4

ACQUIRING PROTOTYPICAL FUNCTIONS FOR PHYSICAL OBJECTS

Humans are a creative species. New objects are invented by people every day, and most are created for a reason. Knives were created for cutting, bicycles were created for transportation, and telephones were created for communication. Some objects can perform multiple functions (e.g., smart phones) and humans are also creative at finding secondary uses for objects (e.g., heavy objects are often used as makeshift paperweights). But when we mention physical objects in conversation or in writing, people generally infer that the object will be used in the most prototypical way (unless they are told otherwise), which we refer to as the *prototypical function* of the object. In this chapter, we will tackle the problem of how to learn this type of knowledge.

The *prototypical function* of human-made physical artifacts is a kind of commonsense knowledge that often plays a role in natural language understanding. Consider the following examples of inferences that arise from physical artifacts.

Example 1

a) He killed the mayor with a *gun*.
b) He killed the mayor with a *knife*.
c) He killed the mayor with a *bomb*.

Example 1 describes a killing with three different types of instruments. Most readers would assume that a) describes a shooting, b) describes a stabbing, and c) describes an explosion. But exactly how each instrument was used is implicit. We make different inferences about how they were used based on our knowledge of the objects.

Example 2 illustrates how we infer different actions based on the object when the main action is elided (i.e., “finished” means that some action has ended but the action itself

Example 2

a) She finished the *cigarette*.
b) She finished the *puzzle*.
c) She finished the *movie*.

is implicit). Most people would assume that the cigarette was smoked, the puzzle was solved, and the movie was watched.

Example 3

a) She put the cake in the *box*.
b) She put the cake in the *oven*.
c) She put the cake in the *refrigerator*.

Example 3 illustrates second-order inferences that can follow from a sentence. The verb “put” means that the cake was placed somewhere, but the object of “in” leads to different inferences about intention. Putting a cake in an oven implies that it will be baked, but putting a cake in a refrigerator implies that it will be cooled.

Example 4

a) He ordered a *taxi*.
b) He ordered a *pizza*.
c) He ordered a *t-shirt*.

Example 4 reveals inferences about motivations and future plans. If someone orders a taxi then we infer that they need transportation, if they order a pizza then we expect they will eat it, and if they order a t-shirt then we assume it will be worn.

We believe that it is essential for NLP systems to “read between the lines” and make the same types of inferences that people do when reading these sentences. The goal of our research is to explore methods for learning the prototypical functions of human-made physical artifacts so that future NLP systems can benefit from this knowledge. First, we define a new NLP task to associate physical objects with frames from FrameNet as a canonical representation for their prototypical function. We introduce a gold standard data set of 938 physical artifacts that have each been labeled with a frame that represents its prototypical function based on human judgements. Second, we evaluate baseline models to assess how well existing resources and simple methods perform on this task. Third, we present transformer-based models for this task that exploit both masked sentence patterns and the definitions of physical artifacts and frames. Experiments show that our best model yields substantially better results than the baseline methods.

4.1 Motivation

This work was motivated by observing sentences that mention physical objects and realizing that we often infer a richer meaning for these sentences than what they explicitly state. We came to appreciate that the prototypical function of an object was the basis for many of our inferences, but we also recognized that not all objects have a prototypical function. In particular, naturally occurring objects rarely have a prototypical function (e.g., *rock*, *snake*). In contrast, human-made physical objects usually do have a prototypical function because they were created for a purpose. Consequently, we limited the scope of our work to human-made artifacts. Of course, some objects are commonly used for multiple purposes, but in most cases there seems to be one use that is dominant, so for the sake of tractability we decided to assign a single (most) prototypical function to each artifact for this research. We had initially planned to include food items, but many foods are also naturally occurring plants or animals (e.g., *watermelon*, *shrimp*) so we omitted them. It may be worth re-examining these limitations in future work.

Another key decision that we had to make was how to represent the prototypical functions. Some recent work on commonsense knowledge acquisition has opted to generate words and phrases as expressions of a relation, such as ConceptNet [159] and ATOMIC [146]. As an example, ConceptNet includes a relation called UsedFor that lists the following phrases as uses for a knife: *stabbing, butter, cutting food, carving wood, slicing, boning*.

We chose to adopt a different approach. First, we wanted a canonical representation for each type of function that represents a general concept, rather than a list of phrases. This approach naturally captures clusters of objects (i.e., those assigned to the same frame) and avoids evaluation issues arising from differing phrases that may be learned for similar objects (e.g., *cut* vs. *carve* vs. *slice*). Second, we did not want to reinvent the wheel and develop a new taxonomy of action types ourselves. For these reasons, we chose to use the semantic frames in FrameNet as a canonical representation for our prototypical functions.

Although FrameNet is not perfect nor complete, it contains many of the actions that we needed. Overall, it serves as an appropriate platform for our work.

4.2 Dataset Creation

4.2.1 Artifact Selection

As explained in Section 4.1, our work focuses on artifacts that are 1) physical objects and 2) created by people. To acquire a list of objects that meet these criteria, we extracted all terms in synsets that are descendants of the *artifact.n.01* synset[1] in WordNet [104]. We then removed a term from the list if the artifact sense was not its first sense definition.[2] This process produced 8,822 entries, many of which met our criteria except that the list still contained a lot of abstract terms (e.g., *vocabulary, modernism*).

To address this issue, we turned to Brysbaert et al. [21] which presents concreteness ratings based on crowd sourcing for 37,058 English words and 2,896 two-word expressions. They used a 5-point rating scale ranging from abstract to concrete, so we extracted words with the part-of-speech "noun" and a rating ≥ 4.5, which produced a list of 3,462 concrete nouns. We then intersected this list with the terms extracted from WordNet, producing a set of 1,017 concrete physical artifacts.

4.2.2 Frame Selection

FrameNet 1.7 contains 1,221 frame definitions. However, not all of them are suitable for representing typical uses of physical artifacts, which should be actions that involve a physical object. For example, some frames are intended for abstract nominal categories (e.g., *Calendric_unit* for temporal terms), high-level abstractions (e.g., *Intentionally_act* which sits above more specific frames), and events or states that are not typically associated with physical artifacts (e.g., *Judgement*).

To focus on an appropriate subset of frames, we manually selected 42 frames in FrameNet that represent actions that are common functions of human-made physical artifacts. We intentionally didn't select frames that categorize nouns in a general way. For example,

[1] Except we removed synsets for buildings and roads.

[2] Because the first sense definition in WordNet usually, though not always, represents the most common meaning.

FrameNet contains an *Artifact* frame that includes *oven*, *phone* and *wheel* as its lexical units. This frame only serves to identify terms that represent physical objects, and we wanted frames that represent a function. The list of frames that we used is shown in Table 4.1 along with the frequency with which they occur in our gold standard data set, as described in the next section.

4.2.3 Human Annotation

To create a gold standard data set with frame assignments for the physical artifacts, we recruited 3 human annotators. We presented the annotators with the WordNet definition for each term and asked them to select one frame that captures the most prototypical use for the artifact. In addition to the 42 function frames, we also gave them a *None* option if none of the frames was a good match, and a *Not an artifact* option if the term was not in fact a human-made physical artifact (because our list extracted from WordNet and Brysbaert et al. [21] was not perfect). To prepare the annotators, we asked them to read the definitions of all the frames beforehand and we gave them detailed annotation guidelines to familiarize them with the task.[3] We randomly sorted the artifacts before presenting them to the annotators.

Table 4.1: Frames for prototypical functions of physical artifacts. The frequency with which they occur in our gold standard data set is shown in parentheses.

Artifact Function Frames		
Wearing (145)	Light_movement (16)	Hunting (8)
Containing (76)	Building (15)	Cause_fluidic_motion (6)
Self_motion (69)	Dimension (15)	Eclipse (5)
Protecting (52)	Removing (14)	Inhibit_movement (5)
Supporting (49)	Closure (13)	Performing_arts (5)
Cause_harm (48)	Competition (13)	Setting_fire (5)
Perception_experience (44)	Create_representation (13)	Cause_to_fragment (4)
Make_noise (37)	Bringing (12)	Education_teaching (3)
Cause_motion (24)	Sleep (12)	Excreting (3)
Cutting (19)	Text_creation (12)	Cause_to_be_dry (2)
Cooking_creation (18)	Attaching (11)	Agriculture (1)
Ingestion (18)	Contacting (10)	Commercial_transaction (1)
Reading_activity (17)	Cure (9)	Residence (1)
Grooming (16)	Cause_temperature_change (8)	Rite (1)

[3] The annotation guidelines can be found in Appendix C.

When the annotations were finished, we measured the pair-wise inter-annotator agreement (IAA) using Cohen's kappa. The IAA scores were 0.75, 0.72 and 0.69, with an average of $\kappa = 0.72$. Given the difficulty of this task (44 possible labels), we felt that the human agreement was relatively good.

Finally, we created the gold standard data set[4] by using the majority label from the three human annotators. There were 72 artifacts with no majority label (i.e., the annotators assigned 3 different labels), and 7 terms with the majority label *Not an artifact*, so we discarded these 79 terms. Consequently, our gold standard data set contains 938 physical artifacts that are each labeled with a frame representing its most prototypical function, or labeled as *None* when none of our 42 frames was appropriate.[5] Table 4.2 shows the 10 most frequently assigned frames and a few examples of artifacts assigned to each frame.

4.3 Frame Identification

We have introduced the physical objects and using frames to represent their prototypical functions. Now the task is formed as, given a physical object, our system should predict the most appropriate frame that describes the most typical way people use this object. This goal can be aligned with a standard NLP task, **frame identification**. In this section, we will present a model for frame identification, which will later be applied as one component for

Table 4.2: Examples of artifacts for the top 10 frames.

Frame	Artifact Examples
Wearing	*hat*, *shirt*
Containing	*basket*, *luggage*
Self_motion	*bicycle*, *yacht*
Protecting	*armor*, *helmet*
Supporting	*chair*, *scaffolding*
Cause_harm	*cannon*, *spear*
Perception_exp	*earphone*, *eyeglass*
Make_noise	*bell*, *violin*
Cause_motion	*engine*, *propeller*
Cutting	*knife*, *scissors*

[4] The data set is available at: https://github.com/tyjiangU/physical_artifacts_function

[5] 83 terms were assigned to the *None* category.

learning the prototypical function.

Research on *frame semantics* has grown within the fields of natural language processing and cognitive science since the 1970s as the study of how we associate words and phrases with cognitive structures called *frames*, which characterize a small abstract scene or situation [51, 52]. The Berkeley FrameNet project [5] provides an online lexical database for frame semantics together with a corpus of annotated documents. Frame semantic parsing is the task of automatically extracting frame semantic structures from sentences. The process typically consists of three steps: *target identification*, which identifies frame-evoking predicates in the sentence; *frame identification*, which identifies the evoked frame for each target; and *argument identification*, which identifies arguments of a frame and labels them with semantic roles (frame elements). In this work, we focus on the frame identification problem.

FrameNet 1.7 contains more than 13,000 lexical units (a word lemma with a sense), each associated with a semantic frame. A polysemous word is associated with multiple lexical units (one for each sense), and is therefore linked to multiple frames. The frame identification task requires a system to identify the most relevant frame for a target word or phrase based on its sentence context. Here is an example:

`The pandemic has` *`sparked`* `a lot of problems for the economy.`

Given the target word *sparked*, the goal is to determine which frame should be triggered. The word lemma *spark* has two senses in FrameNet: “with obj. ignite” and “provide the stimulus for”. The former sense is associated with the *Setting_fire* frame and the latter one is associated with the *Cause_to_start* frame. The *Setting_fire* frame is defined as “this frame describes the creation of a flame...”, and the *Cause_to_start* frame is defined as “a cause, animate or inanimate, causes a process, the effect, to begin”. So *Cause_to_start* is the correct frame for this sentence.

Previous work has shown the success of using feature engineering with linear classification models [74] and discriminative probabilistic models [38], which were later improved by applying distributed word representations and deep neural network models [65]. Syntactic information, typically dependency paths, has consistently played an important role in frame identification [39, 120].

This work is motivated by the rich lexicographic information about frames and lexical units provided by the FrameNet database, which has not been fully utilized for the frame identification task. Recent advances in large pre-trained transformer models [43] have demonstrated the ability to capture semantic meaning in dictionary definitions for the related problem of word sense disambiguation [12, 68].

Our model uses the definitions of frames and lexical units in FrameNet as a source of knowledge to help assess the semantic coherence between the target word and candidate frames. Specifically, we utilize the contextual embeddings produced by the BERT [43] model to determine if a candidate lexical unit and frame express the same meaning as the target word in the given context. Our model achieves state-of-the-art performance on two FrameNet datasets and a FrameNet-annotated dataset based on Yahoo! Answers. Our code is open-source and available online.[6]

4.3.1 Identify the Appropriate Frame

Given a sentence and a target word or phrase, the frame identification task assigns the most relevant frame to the target according to the sentence context. Figure 4.1 shows the framework of our model called FIDO (**F**rame **I**dentification with **D**efiniti**O**ns). Our system takes the sentence and the definitions of associated lexical units (senses) and their frames as input to the BERT model, as indicated by the green blocks. Each green block represents the target word in the sentence, one of its senses, and that sense's associated frame in FrameNet. Then we use the output vectors to produce a probability distribution over all

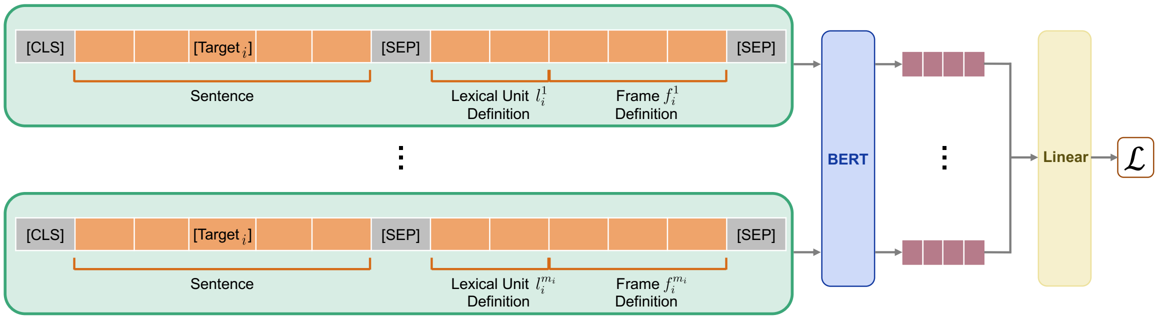


Figure 4.1: Overview of the FIDO architecture. Each green block represents a different candidate pair (lexical unit, frame) for the same Target$_i$.

[6] https://github.com/tyjiangU/fido

of the candidate frames. We select the frame with the maximum probability as the answer.

4.3.1.1 Notation

We denote the ith example ($i = 1, 2, ..., n$) consisting of a sentence and designated target word or phrase as $\langle s, t\rangle_i$, its correct frame as f_i^*, the set of lexical units associated with the target as $l_i^1, l_i^2, ..., l_i^{m_i}$, and their corresponding frames as $f_i^1, f_i^2, ..., f_i^{m_i}$ (f_i^* is among them). We seek to estimate the probability of the jth frame being the correct frame by:

$$\Pr(f_i^j | \langle s, t\rangle_i) = \frac{\exp(g(\langle s, t\rangle_i, f_i^j))}{\sum_{k=1}^{m_i} \exp(g(\langle s, t\rangle_i, f_i^k))} \tag{4.1}$$

where $g(\cdot)$ is a function produced by our model for scoring the assignment of a frame to the sentence and target. We use negative log likelihood as our loss function:

$$\mathcal{L} = -\sum_{i=1}^{n} \log \Pr(f_i^* | \langle s, t\rangle_i) \tag{4.2}$$

where n is the total number of training examples.

4.3.1.2 Modeling

FrameNet provides unique definitions for each lexical unit (LU) and frame. A LU is a pairing of a word lemma and a meaning (sense). Determining the correct LU (sense) uniquely determines the correct frame because each sense of a polysemous word is linked to a different frame. For example, the word *cut* can trigger different frames depending on its meaning (the definition sentences follow the bold lexical unit or frame names), as shown in Table 4.3.

We use the BERT [43] model as the base of our architecture to produce the function $g(\cdot)$ as described in Eq (4.1). For each target, first we extract LUs from FrameNet that have

Table 4.3: Examples of lexical units and their associated frames.

Lexical Unit	Associated Frame
cut.n: the way or style in which a garment or the hair is cut	**Hair_configuration**: temporary or permanent styles and configurations of hair
cut.v: divide into pieces with a knife or other sharp implement	**Cutting**: an agent cuts an item into pieces using an instrument

the same lemma and their corresponding frames to form a set of candidate (LU, Frame) pairs. Our goal is to predict whether the target in the sentence has the same meaning as the definitions of a candidate LU and its associated frame.

As input to the BERT model, we use the sentence as the first sequence and concatenate a LU definition and frame definition as the second sequence. Each definition starts with the LU name or frame name and a colon, followed by the definition description.

Instead of using the output vector of the `[CLS]` token as is typical, we use the last hidden vector of the target word as output (if there is more than one token, we only use the first one). By passing the output vector through a linear layer, we then get a score for assigning a candidate frame to the sentence and target. Finally the scores for all candidate frames are passed through the softmax function to get the probabilities in Eq (4.1).

4.3.2 Evaluation

4.3.2.1 Datasets

• **FrameNet:** To compare FIDO with previous systems, we evaluate our model on FrameNet (FN) 1.5 using the same train/dev/test data split as Das et al. [39]. We also evaluate our model on FN 1.7 which has been available since 2016 and contains nearly 20% more gold annotated data than FN 1.5. We use the same data split as Swayamdipta et al. [164] for FN 1.7. Table 4.4 shows the number of examples in each split.

• **YAGS:** YAGS [63] is a FrameNet-annotated test set based on question answering data from Yahoo! Answers,[7] a community-driven question-and-answer forum. The annotations are based on FN 1.5. We train on FN 1.5 and evaluate on the YAGS test set to compare results with Hartmann et al. [63].

Table 4.4: Dataset sizes.

	FN 1.5	FN 1.7	YAGS
Train	15,017	19,391	-
Dev	4,463	2,272	1,000
Test	4,457	6,714	2,093

[7] https://webscope.sandbox.yahoo.com/

4.3.2.2 Training Details

We use the pre-trained uncased BERT$_{\text{BASE}}$ model with the same settings as Devlin et al. [43] and fine-tune on our training data. We set the max sequence length as 300, batch size as 16, learning rate started at 2e-5, and train for 5 epochs. All reported results are averaged over 3 runs with random seeds.

4.3.2.3 Results

Table 4.5 compares our model with previous methods on the FN 1.5 dataset. Hermann et al. [65], Hartmann et al. [63], and Open-SESAME [164] use distributed representations and syntactic features with neural networks. Botschen et al. [18] extends Hartmann et al. [63] with visual embeddings. Yang and Mitchell [188] integrates a sequential and relational network for joint learning. Peng et al. [120] has achieved the best prior results on frame identification using a multitask approach to learn semantic parsers from disjoint corpora. It is worth noting that besides the FN 1.5 training set, they also use 153,952 exemplar sentences for training, which is more than 10 times the size of our training data. FIDO achieves better performance than all of the prior systems.

Table 4.6 shows our results compared to Peng et al. [120] on FN 1.7. FIDO achieves

Table 4.5: Accuracy on FN 1.5.

Model	Accuracy
Hermann et al. [65]	88.4
Hartmann et al. [63]	87.6
Yang and Mitchell [188]	88.2
Open-SESAME (2017)	86.9
Botschen et al. [18]	88.8
Peng et al. [120]	90.0
FIDO	**91.3**

Table 4.6: Accuracy on FN 1.7 and YAGS.

Dataset	Model	Accuracy
FN 1.7	Peng et al. [120]	89.1
	FIDO	**92.1**
YAGS	Hartmann et al. [63]	62.5
	FIDO	**70.5**

a 3.0% absolute accuracy gain on this data set. The YAGS data set [63] contains *unknown* targets that do not have related LUs in FN 1.5 and also *unlinked* targets (i.e., the provided gold frame does not belong to the set of frames associated with this target in FN). Our model is not able to make a correct prediction for these cases based on its design. There are 122 unknown or unlinked targets in the test set, on which our model will get a zero score. Despite this limitation, our model still outperforms Hartmann et al. [63], which demonstrates its ability to generalize across text genres.

4.3.2.4 Analysis

We performed an ablation study to assess the contributions of each part of our model. In Table 4.7, the first row shows the results for our complete FIDO model. Rows 2-3 show results when using only the definitions of frames (FRdef only) or LUs (LUdef only). We see that the frame definitions contribute the most to performance. Using the LU definitions alone on FN 1.7 also achieves quite good results. But combining both definitions together yields better results than either one alone.

In order to tease apart the impact of the definitions from the impact of BERT, we did an experiment replacing each definition simply with the name of the frame or LU. These results appear in the FIDO (NO def) row. Removing the definitions results in a large performance drop. The definitions clearly play a major role.

In the bottom row, we show the results of experiments using the output vector of the `[CLS]` token (all other settings the same), which did not perform as well as using the target token. This is not surprising as `[CLS]` aggregates the entire sequence representation rather than focusing on the target.

Previous work also reported accuracy on ambiguous cases (i.e., when the target word is

Table 4.7: Ablation study on FN 1.5 and FN 1.7.

Model	FN 1.5	FN 1.7
FIDO	**91.3**	**92.1**
FIDO (FRdef only)	90.1	91.1
FIDO (LUdef only)	88.9	90.7
FIDO (NO def)	80.3	79.4
FIDO (CLS)	89.3	90.5

associated with multiple frames), which more directly shows the model's ability to disambiguate frames. However, the set of ambiguous targets is different across papers. To avoid comparing apples and oranges, we report accuracy on two different sets of ambiguous targets. In Table 4.8, the **Amb1** column follows Peng et al. [120], which uses the gold LU's part-of-speech (POS) tag to form the candidate frame list. In this setting, if a target has just one sense when its POS is known, it is not considered to be ambiguous. Our model outperforms Peng et al. [120] on both FN 1.5 and FN 1.7 datasets. The **Amb2** column shows the accuracy of ambiguous targets using only the lemma of the target (i.e., not relying on gold POS tags). We encourage future work to articulate which setting is used.

We also analyzed whether unseen frames and unseen targets were a major source of errors for our model. On FN 1.7, our FIDO model achieved 92.1% accuracy, so it mislabeled 7.9% of the test cases. We found that 1.4% of the test cases were mislabeled and had an unseen frame (i.e., the gold frame was not seen with the target in the training data), and 0.52% of the test cases were mislabeled and had an unseen target (i.e., the target was not seen in the training data). Therefore only about 1/4 of the FIDO errors were due to unseen frames and unseen targets. We conclude that even for frames and targets that appear in the training data, there is still substantial room for improvement on this task.

4.3.3 Summary

In this section, we tackled the frame identification problem by assessing the semantic coherence between the meaning of a target word in a sentence, and a candidate frame. Our model FIDO exploits the frame and lexical unit definitions provided by FrameNet and a pre-trained transformer model to generate semantic representations. The experiments show that this model achieves better performance than previous systems on two versions of FrameNet data and the YAGS dataset. This relatively simple architecture that brings

Table 4.8: Accuracy on ambiguous cases.

Dataset	Model	Amb1	Amb2
FN 1.5	Peng et al. [120]	78.0	-
	FIDO	**81.0**	83.6
FN 1.7	Peng et al. [120]	77.5	-
	FIDO	**83.8**	85.9

together pre-trained language models with frame and sense definitions can produce a highly effective system for frame identification. In the next section, we will show how FIDO can help identify the prototypical functions.

4.4 Learning Prototypical Functions

In this section, we will introduce several approaches for learning the prototypical functions of human-made physical artifacts. To assess the difficulty of this task, we first present baseline models that 1) exploit information extracted from existing knowledge bases and 2) use co-occurrence information extracted from a text corpus. Next, we explore methods that use large neural language models. We describe a method that uses masked pattern predictions, and then present models that also incorporate artifact sense definitions and frame definitions.

4.4.1 Notation

We model our task as a multiclass classification problem. The artifacts and frames are denoted as a_i $(i = 1..m)$ and f_j $(j = 1..n)$. The task is to select the f_j that represents the most prototypical use for an artifact a_i. We will denote the set of lexical units for f_j in FrameNet as $LU_j = \{l_k | l_k \text{ evokes } f_j\}$.[8]

4.4.2 ConceptNet and COMET Baselines

ConceptNet [159] is a well-known commonsense knowledge resource that contains a UsedFor relation, which is potentially relevant to our task (though it should be noted that an object can be used in ways that are not prototypical, so our task of identifying the *prototypical* use is not exactly the same). COMET [16] is a framework that was trained on ConceptNet with the goal of improving upon its coverage. Our first experiments apply these resources to see how effective they can be for this task.

For each artifact in ConceptNet, we extract the first word from each phrase listed under its UsedFor relation. These are typically verbs that describe an action although sometimes they are nouns. For COMET, we use its *beam-10* setting to generate 10 phrases of the UsedFor relation for each artifact.

[8] We merged lexical units from similar frames in FrameNet. See details in Section 4.5.3.

Next, we want to use the extracted words to rank candidate frames. FrameNet defines *lexical units* that can evoke a specific frame. For example, *read* can trigger the *Reading_activity* frame. Suppose our artifact is a *book* and one of the extracted words is *read*, then *Reading_activity* is a candidate frame. We then score each frame based on the overlap between the words extracted from ConceptNet or COMET and the frame's lexical units. Specifically, we define $freq(a_i, w)$ as the count of a word w occurring in the UsedFor relation of artifact a_i, and $I(w, f_j) = 1$ if $w \in LU_j$ otherwise 0. Then our score for f_j is defined as:

$$S_{cn}(a_i, f_j) = \sum_{w \epsilon W} freq(a_i, w) * I(w, f_j), \tag{4.3}$$

where W is the set of extracted words. Finally, for each a_i, we select $f_{j'}$ such that $j' = \arg\max_j S_{cn}(a_i, f_j)$ as its prototypical function. If $S_{cn}(a_i, f'_j)$ equals zero, then we predict *None*.

4.4.3 Co-occurrence Baseline

An intuitive idea for potentially learning common functions associated with physical artifacts is to extract verbs that frequently co-occur with the artifact in a large text corpus. We assume that if a verb frequently co-occurs with an artifact, then the frames associated with the verb are plausible candidates for the artifact's prototypical function.

For this approach, we created 3 dependency parse patterns to extract <noun, verb> pairs, as depicted in Figure 4.2. The physical object is the noun represented by **N**. The activity is a verb (with an appended particle if one exists) represented by **V**. We included the verb-dobj pattern because some artifacts and their functions are expressed in this way, such as "*read book*" or "*wear jacket*". We used spaCy[9] to parse the whole English Wikipedia corpus (as of Feb 20, 2020) and extracted over 3.8 million <N, V> pairs (305,055 distinct

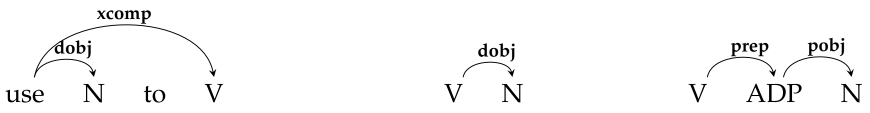


Figure 4.2: Dependency patterns used for co-occurrence.

[9] https://spacy.io/

pairs) for our 938 artifacts. We define the function $freq(a_i, v)$ as the co-occurrence count of artifact a_i and verb v in the corpus. Then we apply the same method described in Section 4.4.2 to assign a score to each frame based on the extracted verbs and select the best frame.

4.4.4 Masked Language Model (MLM) Baseline

Co-occurrence in text is a strong signal of correlation. But an activity that is highly correlated with an artifact may not be its prototypical use. For example, *cut* frequently co-occurs with *rope*, but the purpose of a *rope* is not to be *cut* – its prototypical use is for attaching things.

Recent work has successfully used masked language models to learn commonsense knowledge [41], so we explored whether masked language models could be beneficial for our task. We use the BERT [43] masked language model to get prediction scores for every (a_i, l_k) pair, where a_i is one of our physical artifacts and l_k is a lexical unit linked to one of our 42 candidate frames. We defined 6 sentence templates that represent expressions describing what an object is used for, which are shown below. The first blank space is for artifact a_i and the second blank space is for action l_k:

```
(1) __ can be used to __ .
(2) I used __ to __ .
(3) __ can be used for __ .
(4) I used __ for __ .
(5) The purpose of __ is to __ .
(6) If I had __ , I could __ .
```

Next, we produced a probability distribution over all of the lexical units based on the second blank position. Specifically, for the t-th sentence template s_t, we obtain $Pr(l_k|s_t, a_i)$ by masking only the second blank space (a_i is inserted into the first blank) and we obtain $Pr(l_k|s_t)$ by masking both blank space. Then we define the score of l_k as the typical use of artifact a_i based on the t-th template as:

$$U(a_i, l_k, s_t) = \log Pr(l_k|s_t, a_i) - \log Pr(l_k|s_t). \tag{4.4}$$

The score $U(a_i, l_k)$ using all templates is computed as: $U(a_i, l_k) = \frac{1}{t}\sum_t U(a_i, l_k, s_t)$.

Finally, we define the score for f_j being the prototypical function for a_i as:

$$S_{mlm}(a_i, f_j) = \sum_{l_k \in LU_j} U(a_i, l_k). \tag{4.5}$$

We select $f_{j'}$ where $j' = \arg\max_j S_{mlm}(a_i, f_j)$ as the best frame. If $S_{mlm}(a_i, f'_j) \leq 0$, we predict *None*.

4.4.5 Learning from Masked Patterns

Our MLM baseline uses the discrete output of the masked language model (i.e., the prediction tokens from the vocabulary and their scores). In order to take advantage of a language model's fine-tuning capability, we use the same architecture as described in Section 4.4.4, except that instead of using the predicted lexical units and their probability $Pr(l_k|s_t, a_i)$, we retrieve the last hidden state vector for the `[MASK]` token as output. Since there are 6 masked templates, we have 6 output vectors for each artifact a_i. We compute the average of these vectors and pass it through a linear layer and a softmax layer to produce a probability distribution over all candidate frames plus *None*.

Figure 4.3 shows the overview of this architecture, which we will call the PF$_{mask}$ model. We will refer to the final score for artifact a_i with respect to frame f_j as $S_{mask}(a_i, f_j)$. The loss function is defined as:

$$\mathcal{L} = -\sum_{i=1}^{n} \log S_{mask}(a_i, f_{j^*}), \tag{4.6}$$

where f_{j^*} is the gold label for a_i.

4.4.6 Learning from Definitions

The challenge for our task is obtaining information about the intended function of a physical artifact. We observed that this information is often described in the dictionary

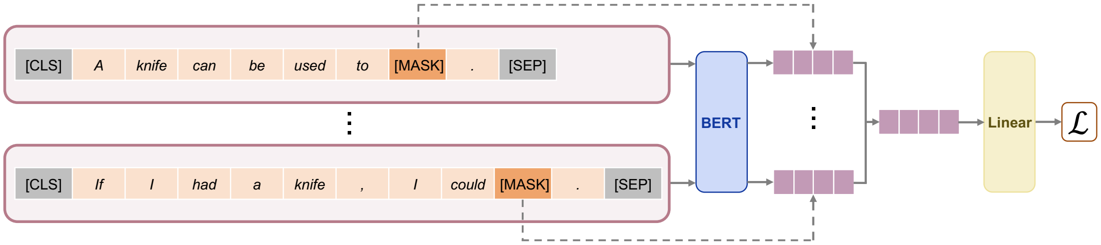


Figure 4.3: Overview of the PF$_{mask}$ model. Each pink block that is fed into BERT represents a sentence template for a given artifact.

definition of an artifact, although it can be expressed in many different ways. For example, the first sense definition in WordNet for *knife* is *"edge tool used as a cutting instrument..."*, and for *bus* it is *"a vehicle carrying many passengers..."*. The definition often provides a short and precise sentence that describes what the artifact is as well as what it is typically used for.

FrameNet also provides a definition for each frame. For example, the definition of the *Cutting* frame is *"An Agent cuts a Item into Pieces using an Instrument"*. In the previous section, we exploited both frame and lexical unit definitions for the frame identification task in a model FIDO (Jiang and Riloff [71]) that assesses the semantic coherence between the meaning of a target word in a sentence and a candidate frame. Here, we hypothesized that a model could potentially learn the semantic relatedness between the definitions of a physical artifact and the frame that describes its typical function.

To investigate this idea, we used the BERT model [43] as the base of our architecture and fine-tuned BERT for our task using both dictionary definitions of artifacts and frame definitions from FrameNet. Figure 4.4 shows the overview of this architecture, which we call the PF$_{def}$ model. Each large green block represents an artifact a_i paired with one of the candidate frames. We encode WordNet's definition of the artifact as the first input sequence and the frame's definition from FrameNet as the second input sequence to BERT. We use the last hidden vector of the `[CLS]` token as the output. For each artifact a_i, we have $n+1$ such pairs where n is the number of candidate frames and 1 refers to the *None* option. On top of BERT's output, we apply a linear and a softmax layer to produce a probability distribution over all candidate frames. We will refer to the final score for artifact a_i with respect to frame f_j as $S_{def}(a_i, f_j)$. The loss function is defined as:

$$\mathcal{L} = -\sum_{i=1}^{n} \log S_{def}(a_i, f_{j^*}), \tag{4.7}$$

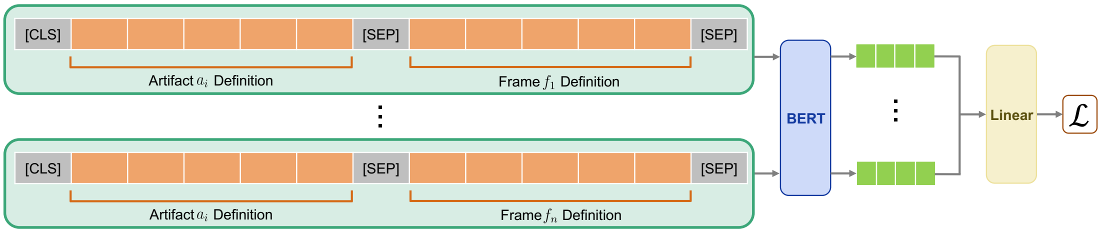


Figure 4.4: Overview of the PF$_{def}$ model. Each green block that is fed into BERT represents an artifact and one of the candidate frames.

where f_{j^*} is the gold label for a_i.

4.4.7 Joint Model

Our final model combines the idea of using both definitions and masked sentence patterns. Figure 4.5 depicts the combined P$_{def+mask}$ model. The left part is the PF$_{def}$ model which estimates the relatedness between artifact and frame definitions. Its output is a matrix of dimension (*# of frames, hidden vector size*). The right part is the PF$_{mask}$ model, which predicts the most probable frame for an artifact using our masked patterns. It produces a single output vector of dimension (*1, hidden vector size*). We broadcast it across the rows to have the same dimension as (*# of frames, hidden vector size*) and then we concatenate the matrices of both models to pass through a linear layer before computing the loss. The model uses fine-tuning to jointly learn all parameters so that information from both models will optimally contribute to the final prediction.

4.5 Evaluation

4.5.1 Experiment Settings

Our gold standard data set contains 938 artifacts that are each paired with one frame that represents its most prototypical use. We set aside 20% (188) of the data as a development set and used 80% (750) as the test set. We evaluated all of the learning models by performing 5-fold cross validation on the test set. We use the pre-trained uncased

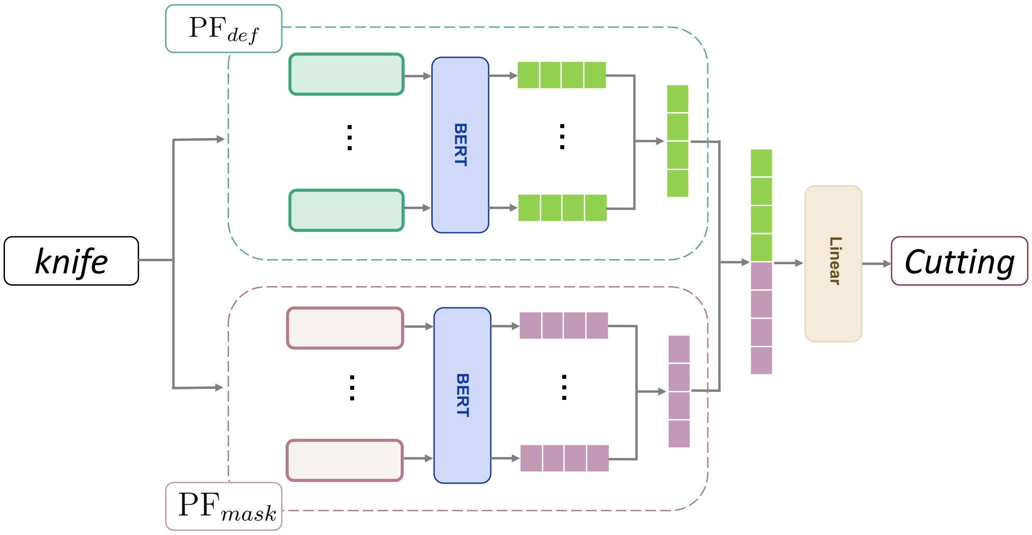


Figure 4.5: Overview of the PF$_{def+mask}$ model.

BERT-base model with the same settings as Devlin et al. [43] and fine-tuned BERT on the training data. We set the max sequence length as 200, batch size as 1, learning rate started at 2e-5, and train for 10 epochs. All reported results are averaged over 3 runs. We report overall accuracy as well as precision, recall and F1 scores macro-averaged over the 43 class labels (42 frames + *None*).

4.5.2 Results

The first four rows in Table 4.9 show the performance of our four baseline methods. ConceptNet and the Co-occurrence model produced the lowest F1 scores. We see that ConceptNet has better precision but low recall because only about 1/3 of the artifacts in our data set has a UsedFor relation defined in ConceptNet. We also tried adding the CapableOf relation, which is defined as what an item can do, but it is even more sparse than UsedFor and combining both relations only marginally increased recall. The performance of COMET shows that COMET does indeed improve upon the coverage of ConceptNet, although it sacrifices some precision. We also tried using the *beam-5* and *greedy* settings of COMET, which produced higher precision but lower recall and F1 scores.

Compared to COMET, the Co-occurrence baseline has higher accuracy but a much lower F1 score. The explanation is that the Co-occurrence model performs much better on frames that are associated with artifacts that are frequently mentioned in the corpus than for frames associated with less frequent artifacts. This is intuitive because, in general, we expect to extract a more representative sample of activities when we have more data. This phenomenon (accuracy much higher than F1) can also be observed in the MLM model which uses a pre-trained language model that learns from large corpora, so it is

Table 4.9: Experimental results for different models.

Model	**Acc**	**Pre**	**Rec**	**F1**
ConceptNet	17.5	33.6	13.5	16.4
Co-occurrence	31.9	24.1	23.9	19.9
COMET	30.7	29.7	35.6	28.2
MLM	42.8	29.5	33.8	28.2
PF$_{mask}$	58.5	35.7	36.5	35.4
PF$_{def}$	74.7	63.5	57.6	59.3
PF$_{def+mask}$	**76.8**	**65.2**	**61.1**	**62.4**

not surprising that Co-occurrence and the MLM model behave similarly. In contrast, ConceptNet and COMET behave more consistently across the set of frames.

The bottom two sections of Table 4.9 show the results for our new models, which were trained specifically for this task. The PF$_{mask}$ model achieves 58.5% accuracy and a 35.4% F1 score, which outperforms all of the baselines. The PF$_{def}$ model performs substantially better, achieving 74.7% accuracy and a 59.3% F1 score. This result demonstrates that the definitions of the artifacts and the frames provide valuable information that a learner can benefit from. The last row shows the performance of the combined model, which performed better than the individual models. This model saw additional gains in both precision and recall, increasing the accuracy from 74.7% to 76.8% and the F1 score from 59.3% to 62.4%.

4.5.3 Analysis

4.5.3.1 Training Size

To understand the degree to which the number of training instances for each frame correlated with performance, we divided the frames into two sets: high frequency frames assigned to ≥ 15 artifacts and low frequency frames assigned to < 15 artifacts. The results are shown in Figure 4.6 with the F1 scores from the PF$_{def+mask}$ model displayed on the Y-axis.

We conclude that frames with more training instances generally showed better performance, so our model would likely further improve given more training data.

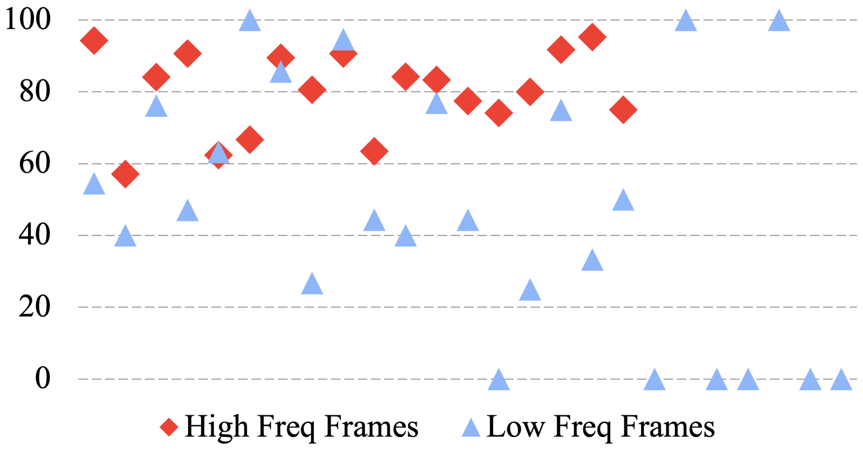


Figure 4.6: F1 scores for high and low frequency frames.

4.5.3.2 Compare PF$_{mask}$ and PF$_{def}$

Table 4.10 shows some examples of output from the PF$_{mask}$ and PF$_{def}$ models to compare their behavior. The correct predictions appear in bold. Both models are correct for example 1. For example 2, only the PF$_{mask}$ model is right, which indicates that the masked pattern can be more useful than the definition sometimes. For examples 3 and 4, PF$_{def}$ was correct and PF$_{mask}$ was wrong. The PF$_{mask}$ model sometimes generates frames representing functions that are true but tangential. For example, beds do support us and helmets are worn, but these functions do not sufficiently characterize the objects (e.g., chairs also support us but are not typically used for sleeping, and jewelry is also worn but not used for protection). For example 5, both models are wrong – the correct frame is *Removing*. Though both are wrong, the PF$_{def}$ model produces a more reasonable answer than the PF$_{mask}$ model.[10] We also observed that the MLM baseline sometimes produces seemingly random answers that are hard to explain.

4.5.3.3 Frames Labeled as None

We investigated the 83 instances that were labeled as *None* to see what kind of artifacts fell into this category. The biggest cluster of related artifacts were 17 types of fabric, such as *linen*, *silk* and *canvas*. FrameNet does not include a frame for materials of this kind, probably because they are an ingredient for making clothes rather than tools themselves. Artifacts like *toy* were also labeled as *None* presumably because toys are used in a general way (for play). This category also included some artifacts not tied to a single prototypical function but commonly used for many purposes (e.g., *computer, laptop*).

Table 4.10: Sample output of PF$_{def}$ and PF$_{mask}$ models. The correct predictions are in bold.

		ID	**Artifact**	**PF$_{mask}$**	**PF$_{def}$**
MASK ✓	DEF ✓	1	scissors	**Cutting**	**Cutting**
MASK ✓	DEF ✗	2	hydrant	**Cause_fluidic_motion**	Cause_temperature_change
MASK ✗	DEF ✓	3	bed	Supporting	**Sleep**
MASK ✗	DEF ✓	4	helmet	Wearing	**Protecting**
MASK ✗	DEF ✗	5	snowplow	Hunting	Self_Motion

[10] In fact, snowplow can also refer to a skiing action, although WordNet does not contain that word sense.

4.5.3.4 Merging Lexical Units

When selecting frames to represent the prototypical functions of physical artifacts, we observed that some frames in FrameNet share similar meanings (e.g., *Reading_activity* and *Reading_perception*) or related functions (e.g., *Create_representation* and *Recording*). However, these frames often have complementary sets of lexical units.

Since our baselines (ConceptNet, COMET, Co-occurrence, and MLM) rely on the lexical units of frames to make predictions, increasing the coverage of lexical units can be beneficial. So we manually clustered frames that share a related definition with our 42 chosen frames and merged their lexical units. Table 4.11 shows the cluster for which the lexical units are merged.

Table 4.12 shows the experimental results for our baselines with and without merging the lexical units (lexical units are only used by the baseline models). The first four rows show results without merging lexical units and the last four rows show results with merged lexical units (same numbers reported in Table 4.9). We can see that merging lexical units from frames in Table 4.11 helps improve both the precision and recall. It also indicates

Table 4.11: Clustered frames. The left column are from our pre-selected frames. The right column are related frames in FrameNet providing complimentary lexical units.

Primary Frame	**Clustered Frames**
Agriculture	Food_gathering, Growing_food, Planting
Attaching	Connectors
Cause_fluidic_motion	Cause_to_be_wet
Cause_harm	Attack, Weapon
Cause_motion	Cause_to_move_in_place
Cause_to_fragment	Grinding
Commercial_transaction	Commerce_buy, Commerce_sell
Competition	Exercising
Containing	Containers
Cooking_creation	Apply_heat
Create_representation	Recording
Cure	Recovery
Hunting	Taking_captive, Trap
Inhibit_movement	Immobilization
Light_movement	Location_of_light
Make_noise	Cause_to_make_noise, Noise_makers
Perception_experience	Perception_active, Cause_to_perceive, Information_display
Reading_activity	Reading_perception
Removing	Emptying
Self_motion	Vehicle, Ride_vehicle, Operate_vehicle
Supporting	Posture
Wearing	Body_decoration, Clothing, Accoutrements, Clothing_parts

Table 4.12: Experimental results with and without merging lexical units.

LU Merge	Model	Acc	Pre	Rec	F1
Before	ConceptNet	15.7	32.2	12.3	14.5
	Co-occurrence	22.5	19.5	17.5	14.8
	COMET	23.9	24.5	28.3	21.4
	MLM	26.1	24.1	28.5	21.4
After	ConceptNet	17.5	33.6	13.5	16.4
	Co-occurrence	31.9	24.1	23.9	19.9
	COMET	30.7	29.7	35.6	28.2
	MLM	42.8	29.5	33.8	28.2

that a more appropriate set of trigger words should potentially benefit the task.

4.6 Related Work

Researchers have known for a long time that commonsense knowledge is essential for natural language understanding [30, 149]. Some of this work specifically argued that commonsense knowledge about physical objects, including functional knowledge, plays an important role in narrative text understanding [22, 83].

These observations have led to considerable work toward constructing commonsense knowledge repositories. The Cyc project [84] built a large ontology of commonsense concepts and facts over many years. More recently, ConceptNet [159] captures commonsense knowledge in the form of predefined relations expressed in natural language words and phrases. It was built from Open Mind Common Sense, a crowd-sourced knowledge project [156], and later enhanced with other sources such as Wiktionary and WordNet [104].

Within the NLP community, a variety of recent projects have focused on trying to acquire different types of commonsense knowledge, such as Forbes and Choi [54], Collell et al. [35], Yang et al. [189] and Event2Mind [138]. Sap et al. [146] presented a crowd-sourced commonsense reasoning data set called ATOMIC that focuses on inferential knowledge related to events, which is organized as if-then relations. Bosselut et al. [16] later proposed COMET, a transformer-based framework for automatic construction of commonsense knowledge bases that was trained from ATOMIC and ConceptNet. Both ConceptNet and COMET include a UsedFor relation that is relevant to our task, and we evaluate their performance on our data set in Section 4.5.

Of relevance to this work, Jiang and Riloff [73] (Chapter 3) learned the prototypical "functions" of locations by identifying activities that represent a prototypical reason why people go to a location. For example, people go to restaurants to eat, airports to catch a flight, and churches to pray. We referred to the associated activity as a prototypical goal activity and presented a semi-supervised method to iteratively learn the goal activities.

Our work is also related to frame semantics, which studies how we associate words and phrases with conceptual structures called frames [51], which characterize an abstract scene or situation. The Berkeley FrameNet project [5, 141] provides an online lexical database for frame semantics and a corpus of annotated documents. Several efforts have enhanced FrameNet by mapping it to other lexicons, such as WordNet, PropBank and VerbNet [50, 114, 152]. Pavlick et al. [119] increased the lexical coverage of FrameNet through automatic paraphrasing and manual verification. Yatskar et al. [191] introduced situation recognition, which is the problem of producing a concise summary of the situation that an image depicts. Similar to our work, they selected a subset of frames from FrameNet to represent possible situations depicted in an image. Our work uses a subset of frames from FrameNet to represent the prototypical functions for human-made physical artifacts.

There has been considerable work on the frame identification problem with respect to FrameNet, especially since the SemEval 2007 shared task [6]. Johansson and Nugues [74] used a SVM classifier to disambiguate frames with hand-crafted features. Das et al. [38] applied feature-based discriminative probabilistic (log-linear) models for frame identification. Hermann et al. [65] presented a method using distributed representations of predicates and their syntactic context by mapping input representations and frame representations to a common latent space using the WSABIE algorithm [180]. Hartmann et al. [63] built a simplified model based on Hermann et al. [65] and achieved comparable results. They also released a new FrameNet-annotated test set based on user-generated web text from Yahoo! Answers. Yang and Mitchell [188] integrated a bidirectional LSTM neural network and a relational network to jointly decode frames.

More recently, Botschen et al. [18] brought in multimodal representations grounded in images to improve frame identification. Peng et al. [120] proposed a joint inference formulation that learns semantic parsers from multiple datasets.

In contrast to the previous models, our FIDO model does not rely on syntactic features. We assess semantic coherence directly from the input sentence and definitions in

FrameNet.

Another line of related work is learning embeddings from dictionary definitions. It has been shown that neural networks can extract semantic information from dictionary definitions [15, 79]. Previous work in word sense disambiguation [12, 68] has demonstrated that providing pre-trained language models with sense definitions (glosses) can be effective. Yong and Torrent [192] also used the sense definitions of lexical units for their research on frame induction. Our FIDO model adopts a similar architecture as Huang et al. [68], but we focus on the frame identification task and we explore the use of both lexical unit and frame definitions for the task.

4.7 Conclusion

We introduced the new task of learning prototypical functions for human-made physical artifacts, and used a subset of frames from FrameNet to represent the set of common functions. We also presented a manually annotated data set of 938 physical artifacts for this task. Our experiments showed that a transformer-based model using both artifact and frame definitions as well as masked pattern predictions outperforms several baseline methods. In future work, we hope to show the value of functional knowledge about objects for sentence-level understanding tasks as well as narrative document understanding.

CHAPTER 5

IDENTIFYING OBJECT USE IN SENTENCES

In the previous chapter, we have argued that physical objects play an important role in daily life. People use them for different purposes; for example, we use knives for cutting, cars for transportation, and books for reading. Most human-made physical artifacts were created for a specific purpose, and that commonsense knowledge about an object's *prototypical function* is essential for natural language understanding. For example, *"she finished the puzzle"* and *"she finished the cigarette"* implicitly refer to different actions associated with puzzles (solving) versus cigarettes (smoking). Similarly, humans interpret *"he used a gun"* as a shooting but *"he used a knife"* as a stabbing based on our knowledge of guns and knives. In this chapter, we want to demonstrate how can we apply the prototypical function in context understanding.

Chapter 4 presented a method to learn the prototypical functions for physical artifacts using pretrained language models, with the goal of producing a commonsense knowledge resource for physical objects. However, an open question is how to apply this knowledge for sentence understanding.

It would be risky to assume that objects are always used in the most typical way because objects can be used in atypical ways too. For example, *"Max used the knife to open the bottle"* probably means that Max popped the top off the bottle with the knife, not that Max cut the bottle. But as we will discuss in Section 5.2, we found that physical artifacts are used in the prototypical way most of the time (96%), so it is a very good assumption.

A much bigger problem for applying knowledge of prototypical functions is that physical objects are often mentioned when they are not used at all! For example, the sentences below mention a knife, but the knife is not being used:

```
(a) He put the knife in the dishwasher.
(b) She found a knife in the woods.
```

(c) `The knife fell off the table.`

(d) `A good pocket knife costs $100.`

In addition, some sentences suggest that a physical object *will be* used, although it has not been used yet. For example, Mary aims to acquire a knife through various actions in the sentences below:

(e) `Mary got a knife from the drawer.`

(f) `Mary asked John for a knife.`

(g) `Mary purchased a chef's knife.`

When reading these sentences, people naturally infer that Mary intends to use the knife, most likely in the typical way (i.e., to cut things). We believe that NLP systems should also make these predictive inferences to "read between the lines" during narrative text understanding. For example, consider the sentence *"The fish was too big for the freezer, so Mary got a knife."*. Human readers would assume that Mary used the knife to cut the fish into smaller pieces, even without any explicit mention of cutting.

We propose a new NLP task, *Object Use Classification*, to classify the usage status of physical objects mentioned in sentences with respect to three categories: *Used*, *Anticipated Use*, and *No Use*. Our first goal is to identify sentences that state or imply that an object was used (*Used*) to enable prototypical function inferences when the action is implicit. Our second goal is to identify sentences that describe actions which suggest someone's probable intent to use the object (*Anticipated Use*) to enable second-order prototypical function inferences. Finally, identifying sentences where there is no use of a mentioned object (*No Use*) is important to recognize when prototypical function inferences should not be applied. We introduce a new object use dataset for this task with gold standard human annotations of sentences that mention physical objects. We found that all three use categories are common: our annotators labeled 45% of the sentences as *Used*, 28% as *Anticipated Use*, and 27% as *No Use*.

We explored several methods to tackle this task. First we applied prompting methods using two large language models to evaluate a zero-shot generalization approach, but the results were mediocre. Next, we fine-tuned a transformer-based model, which yielded much better performance. Finally, we added two data augmentation techniques, synonym replacement and back translation, and also provided exemplar sentences associated with the object's prototypical function frame. Our experimental results show that the complete model achieves good performance for this task.

5.1 Motivation

Actions involving physical objects are often left implicit in natural language. In many cases, these actions do not need to be explicitly stated because they can be easily inferred by people using our knowledge about physical objects. Early NLP research recognized this need for commonsense knowledge about physical objects (e.g., Burstein [22]) and some efforts have been undertaken to compile such knowledge, including ConceptNet [159], which contains a "UsedFor" relation that captures possible uses for an object expressed in natural language, and our previous work introduced in Chapter 4 that learns to associate physical objects with FrameNet frames describing their prototypical uses.

However, a crucial question is when to apply this knowledge in sentence understanding. We claim that NLP systems must be able to distinguish between (1) sentences that mention a physical object and state or imply that the object was or will be used, and (2) sentences that mention a physical object but the object was not used. For example, the sentences *"Mary read the book"* and *"Mary enjoyed the book"* both imply that the book was used (read), but *"Mary dropped the book"* does not mean that the book was used, only that Mary was carrying it. We found that about 73% of sentences that mention a physical object in our data set (see Section 5.2) suggest that the object was (45%) or will be (28%) used. For the other 27% of sentences that mention a physical object, the object was not used at all. Consequently, we argue that an important task for understanding sentences about physical objects is *object use classification*.

A second question relates to the applicability of "prototypical" functions for objects: when an object *is* used, how often is it used in the prototypical way? In our dataset (Section 5.2), we found that when a sentence mentions or implies the use of an object, 96% of these sentences correspond to the prototypical use for the object. Only 4% of these sentences suggest that an object was used in an atypical way. These results indicate that an effective object use classifier can go a long way toward enabling NLP systems not only to infer *whether* an object was used, but also *how* an object was used, even when that action is not explicitly stated.

In this chapter, we tackle the problem of object use classification and define three categories of object use: *Used*, *Anticipated Use*, and *No Use*. In the next section, we define these three categories and explain our motivation for them, and we present a new object use dataset for this task.

5.2 Dataset Creation

Since we are tackling a new task, we created a new *TOUCAN* (**T**extual **O**bject **U**se Cl**A**ssificatio**N**) dataset with gold standard human annotations. Our primary goal was to obtain human judgements for sentences that mention a physical object indicating whether the object was or will be used, or not. But a second goal was to better understand how often objects are used in a prototypical way, as opposed to an atypical way. So we obtained additional human judgements for the sentences in which an object was or will be used and asked the annotators to determine whether the use corresponds to the object's prototypical function. We leveraged the results of prior work that studied physical objects and their prototypical functions so as not to reinvent the wheel:

- **Physical Objects:** We use the list of physical objects introduced in Chapter 4 Section 4.2.3 (Jiang and Riloff [72]). The human-made physical objects were extracted with sense definitions from WordNet [104], and a concreteness dictionary [21] was used to filter out abstract terms. The list contains 938 human-made physical object terms.

- **Sentence:** Then we extracted sentences containing these physical objects from the Spinn3r corpus [24], which consists of 44 million blog posts. In order to get a uniform distribution of different physical objects, we randomly sampled 4 sentences (or fewer if not enough) for each physical object. This produced a set of 2,460 sentences in total.

5.2.1 Human Annotation

5.2.1.1 Object Use Categories

First, we presented two people with a physical object term, a sentence that mentions the object, and the WordNet definition of the object.[1] We asked the annotators to select one

[1] We manually identified the WordNet definition corresponding to the physical object sense of the term.

of these four categories:[2]

1. Used: The sentence describes 1) an action in which the object is/was being used (by the writer or someone else), or 2) an action that directly resulted from the use of the object.

2. Anticipated Use: The sentence states that 1) the object will be used in the future, or 2) implies that someone will presumably use the object.

3. No Use: Neither *Used* nor *Anticipated Use*.

4. Wrong Sense: The given definition of the object term is different from its meaning in the sentence. (This option was provided to flag sentences in which the term's meaning is not its physical object sense. We do not include these sentences in our dataset.)

5.2.1.2 Prototypical Use Annotations

In Chapter 4, we proposed that most human-made physical artifacts have a prototypical function (i.e., the intended purpose of the object). We selected 42 frames[3] from Framenet v1.7 [141] to represent actions that are common functions of physical artifacts. Table 5.1 shows a few physical objects and their prototypical function frames.

To better understand how often objects are used in a prototypical way as opposed to an atypical way, we collected additional human judgements. If an annotator selected

Table 5.1: Examples of objects and their prototypical function frames.

Frame	**Artifact Examples**
Wearing	*hat*, *shirt*
Containing	*basket*, *luggage*
Self_motion	*bicycle*, *yacht*
Protecting	*armor*, *helmet*
Supporting	*chair*, *scaffolding*
Cause_harm	*cannon*, *spear*
Perception_exp	*earphone*, *eyeglass*
Make_noise	*bell*, *violin*
Cause_motion	*engine*, *propeller*
Cutting	*knife*, *scissors*

[2] The annotation guidelines can be found in Appendix D.

[3] See Section 4.2.2 Table 4.1 for all 42 frames.

Used or *Anticipated Use*, we also asked the annotator whether the use of the object most likely corresponds to its prototypical function (based on the gold frame in our dataset). The annotator was shown the prototypical function frame for the object, and asked to select *Yes* or *No* as to whether the frame correctly characterizes the use of the object in the sentence. The last column in Table 5.2 shows the prototypical function frames, followed by the annotated *Yes* (✓) or *No* (✗). For example, in sentence (3) of Table 5.2, a toothpick is typically used to remove food that is stuck between our teeth, but here it is used to hold sausage and olives together so this sentence represents an atypical use for a toothpick.[4]

To prepare the annotators, we provided them with detailed annotation guidelines to familiarize them with the task. Deciding whether the prototypical function frame is correct requires knowledge of FrameNet frames, so we also asked them to read FrameNet's definitions and exemplar sentences for each relevant frame. We randomly shuffled the physical objects before presenting them to the annotators. The pairwise inter-annotator agreement using Cohen's kappa for the 3 object use categories was 0.71, and the simple agreement rate (percentage of agreement) for the Yes/No prototypical function question was 0.92.

To create the final set of gold standard labels, we had the annotators adjudicate their disagreements. This process produced 2,123 sentences annotated with one of the *Used*, *Anticipated Use* or *No Use* label. The **All Cases** row of Table 5.3 shows the distribution of

Table 5.2: A sample of the annotated examples. The physical objects are marked in red. The second column shows the annotated use category. The third column shows the gold prototypical function frame for the object followed by ✓or ✗. The ✓means the frame is consistent with the use of the object in the sentence, otherwise ✗.

Sentence	Use Category	Prototypical Function
(1) We took a speedboat up the river to the village.	Used	Self_motion ✓
(2) He promptly walked over to his mattress and laid down.		Sleep ✓
(3) I had sausage slices wrapped around olives, held together with a toothpick .		Removing ✗
(4) I quickly went to the bathroom and got more ammo .	Anticipated Use	Cause_harm ✓
(5) All my new cookware will be put to use with these new recipes!		Cooking_creation ✓
(6) I got measured for my tuxedo for Dad 's wedding today.		Wearing ✓
(7) Nope , it also hit my left headlight and broke it.	No Use	-
(8) At one point, I saw a high heel shoe .		-
(9) I promptly threw the brochure in a corner to collect dust.		-

[4] One could argue that toothpicks have multiple common functions, but we defined only one prototypical function for each object in their work.

Table 5.3: Annotated data statistics. The "Prototypical" row shows the number of examples in which the prototypical function of the object is consistent with its use in the sentence.

	Used	Anticipated Use	No Use
All Cases	964	583	576
Prototypical Use	935	560	-

the 3 categories. The Used category accounts for 45% of the sentences, with Anticipated Use and No Use each making up about 27% of the data.

The **Prototypical Use** row shows the prototypical use results. The annotators determined that objects were used in their most prototypical way in 97% (935/964) of the Used sentences and in 96% (560/583) of the Anticipated Use sentences. This data suggests that if we had a perfect object use classifier, we could infer *how* an object was used with 96.6% accuracy simply by assuming its prototypical function.

5.3 Object Use Classification

We explored several approaches to tackle this task. We first present a transformer-based model fine-tuned solely on our gold standard training sentences. Then we present a method that takes advantage of two commonly used data augmentation techniques, synonym replacement and back translation. Finally, we also show that our model further benefits from prior knowledge of the object's prototypical function by incorporating the exemplar sentences associated with its function frame.

5.3.1 Task Definition and Base Model

We model our task as a 3-class classification problem. Given a sentence, and an object mentioned in the sentence, the task is to determine if the object has been used, has an anticipated use in the future, or has no stated or implied use.

We build our model based on RoBERTa [94]. For our base model, we use the sentence as the input sequence into the model, and use the last hidden vector representing the object (if there is more than one token, we compute the average of all tokens) as output, and then pass it through a linear classifier to predict the label.

5.3.2 Synonym and Hyponym Replacement

Our physical object list originated from WordNet. To increase the number of training sentences, we created copies of each original training sentence where the object term is replaced by one of its synonyms or hyponyms in WordNet. Specifically, for each object that has a WordNet synset, we first extract all the lemmas belonging to the same synset and also traverse one level down in WordNet's hierarchy to extract the lemmas of its direct hyponyms. For example, the furniture term *sofa* belongs to the synset `sofa.n.01`, which also contains *couch* and *lounge*. Suppose its direct hyponyms are *daybed*, *divan*, and *loveseat*. Then we have a list of 5 new object terms. For each training sentence that mentions a *sofa*, we replace the word *sofa* with its synonyms or hyponyms, generating 5 new training examples with the same label. In general, we use all of the synonyms and up to 5 hyponyms (if there are more than five then we randomly select five).

5.3.3 Back Translation

Back translation [151] is a widely used data augmentation technique, which automatically generates new training examples by translating a sentence to another language and then translating it back, as illustrated in Figure 5.1. The technique aims to produce diverse paraphrases of the original sentence. Its effectiveness has been shown for downstream tasks such as text classification [185] and question answering [95].

Table 5.3 shows that the label distribution in our dataset is roughly 2:1:1. Though

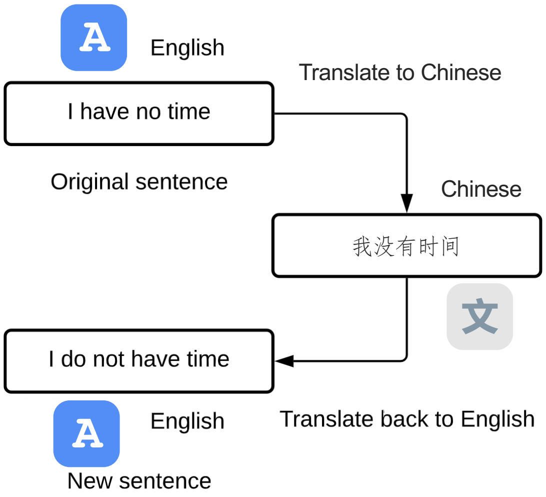


Figure 5.1: An illustration of how back-translation works.[5]

this imbalance reflects the actual distribution of these categories, we hypothesized that the model would perform better with a more balanced distribution of class labels. So we performed back translation on the training examples labeled with *Anticipated Use* and *No Use*, augmenting these categories to be roughly the same size as the *Used* category. For back translation, we use the Helsinki-NLP English to Chinese and Chinese to English transformer-based machine translation system [170].

5.3.4 FrameNet Exemplars

Our annotation results suggested that when objects are used, they are almost always used in the prototypical way. So for each object, we utilize the prototypical function frame in the gold standard dataset as introduced in Chapter 4 as prior knowledge to recognize whether the sentence describes a relevant situation. Specifically, we extract the frame's *exemplar sentences* from FrameNet, which are the annotated sentences associated with a lexical unit that triggers the frame. For example, the *Sleep* frame contains examplar sentences such as *"I was exhausted, and slept for two hours"*, *"Well, better get some shut-eye"*, etc. We concatenate all of the exemplar sentences for the frame as one sequence, then pair it with the sentence containing this object as the input to the RoBERTa model.

5.3.5 Complete Model Architecture

Figure 5.2 shows an overview of the full architecture for our learning process. For simplicity, we call the model TOUCAN as well. First we use synonym/hyponym replacement to augment the original training set. Then we apply back translation to all of the sentences labeled as *Anticipated Use* or *No Use* to generate more sentences, and add them to the training set. When applying back translation, for each sentence, we generate only one new sentence from the translator.

Finally, for each sentence in the training set, we extract the exemplar sentences from FrameNet corresponding to the object's prototypical function frame. The exemplar sentences are concatenated and given to RoBERTa along with the original sentence as input. Then we send the last hidden vector of the object into a linear classifier on top of the

[5] Figure adapted from Beddiar et al. [10].

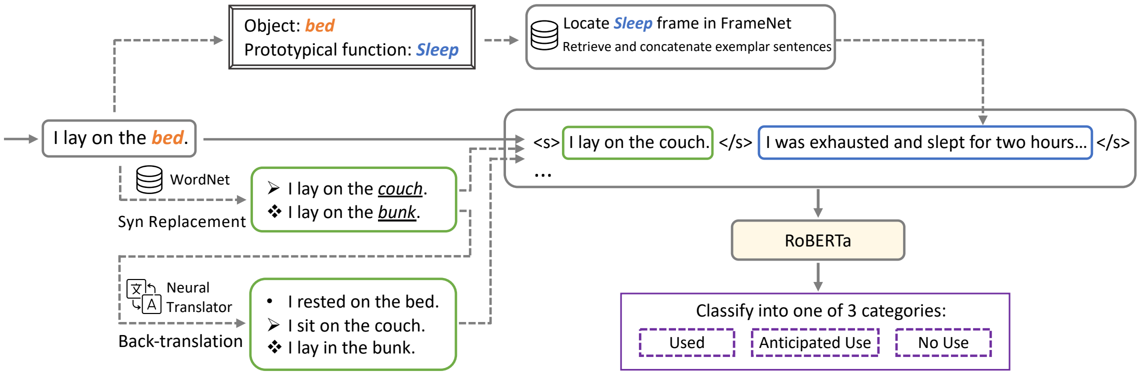


Figure 5.2: An illustration of the TOUCAN object use classification model.

RoBERTa model.[6] If there are multiple tokens, we compute their average.

5.4 Evaluation

We split our gold standard data set into roughly 70% for training, 15% for development and 15% for testing. We also made sure that the objects in the test set do not appear in the training set. Our fine-tuning framework is based on the RoBERTa-base model [94]. For the hyper-parameters, we used a max sequence length of 192, a batch size of 8, learning rate initialized as 2e-5, and train for 15 epochs. Each result is averaged over three runs with different random seeds. We report overall accuracy as well as precision, recall and F1 scores macro-averaged over the 3 classes.

5.4.1 Prompting Baseline

Recent advances in pre-trained language models have demonstrated their ability to attain zero-shot generalization on different downstream tasks [20]. Specifically, prompting has become a widely used technique in natural language processing. It works by recasting NLP tasks in the form of a natural language response to a natural language input. To see how well this approach can work for our task, we explore prompting with two language models: GPT-2 [133] and T0++ [145]. T0++ is an encoder-decoder model that has been trained on a collection of downstream tasks such as question answering and summarization, with multiple prompts per dataset. We cast our problem as a textual entailment task

[6] In rare cases, the object no longer exists in the sentence after back translation. In this case we use the vector of the first token in the sentence.

and use the same set of prompts in [145]. For example, one template is:

```
    Suppose [premise].  Can we infer that ‘‘[hypothesis]’’?  Yes,
or no?
```

Since our task is to distinguish between three different categories (*Used*, *Anticipated Use*, and *No Use*), we created a two-template pipeline to obtain the prediction. As an example, consider the sentence *"John finished the watermelon with the spoon"*, where the spoon is the object in question. Two templates would be generated:

T1: `Suppose John finished the watermelon with the spoon. Can we infer that ‘‘the spoon has been used’’? Yes, or no?`

T2: `Suppose John finished the watermelon with the spoon. Can we infer that ‘‘the spoon will be used in the future’’? Yes, or no?`

If the output for T1 is *Yes*, it means the prediction is *Used*. If the output for T1 is *No* but for T2 is *Yes*, it means the prediction is *Anticipated Use*. Otherwise the prediction is *No Use*. Since the T0++ model has been fine-tuned with the prompt templates, it will always predict *Yes* or *No* as the output. However the GPT-2 model can predict other tokens as the next word. So for GPT-2, we compare the probability score for the *Yes* and *No* tokens and choose the one that is higher. We report results averaged over all templates.

5.4.2 Results

Table 5.4 shows our experimental results. As a baseline, the first row shows that random labeling (assigning each label with the probability of 1/3) achieves 30.0% F1. The next two rows show the results for the prompt-based methods. The GPT-2 model predicts *No* much more frequently than *Yes* and most predictions fall into the *No Use* category. It very rarely predicts *Anticipated Use*. This produces a low F1 score of 18.5%. T0++ also suffers from low recall for *Anticipated Use*, but it is substantially better than GPT-2, achieving 46.0% accuracy and 43.3% F1 score.

The next section of Table 5.4 shows the results for our fine-tuned models. Using the

Table 5.4: Object use results across models.

Model	**Acc**	**Pre**	**Rec**	**F1**
random	31.0	30.5	30.2	30.0
GPT-2	30.1	24.6	31.9	18.5
T0++	46.0	57.7	46.6	43.3
TOUCAN$_{base}$	65.8	64.8	64.1	63.6
+Synonyms	70.0	69.4	68.2	68.3
+BackTrans	72.0	71.6	69.9	69.7
+Syn&BackTrans	71.9	71.5	70.1	70.3
+Exemplar	70.5	69.4	67.5	67.4
TOUCAN	**72.4**	**72.0**	**70.6**	**70.8**

sentence alone (TOUCAN$_{base}$) achieves 65.8% accuracy and 63.6% F1. Each of the following rows adds one new component to the architecture to evaluate its contribution independently (not cumulatively). The *+Synonyms* and *+BackTrans* rows show results for data augmentation using synonym/hyponym replacement and back translation respectively on top of the TOUCAN$_{base}$ model. We see that synonym/hyponym replacement increases the F1 score to 68.3%, and back translation performs even better at 69.7%. When using both synonym/hyponym replacement and back translation (row *+Syn&BackTrans*), the model achieves over 70% F1 score.

The *+Exemplar* row shows the results when giving the sentence as well as FrameNet's exemplar sentences as a sequence pair to RoBERTa. Note that this model requires gold information about an object's prototypical function. Compared to TOUCAN$_{base}$, adding the exemplar sentences increases the F1 score from 63.6% to 67.4%. The last row (TOUCAN) shows the results when combining all of the elements together, which yields the highest accuracy score of 72.4% and highest F1 score of 70.8%.

Table 5.5 shows the performance breakdown for each label. Here we only show T0++ for comparison with the fine-tuned models. A clear difference between T0++ and the fine-tuned models is that T0++ labels far too many instances as *No Use*. The fine-tuned models do a much better job at distinguishing the 3 classes. We can also see that both data augmentation methods help improve recall, especially for the *Used* and *No Use* categories.

Table 5.5: Results breakdown for each label.

	Use			Anticipated Use			No Use		
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
T0++	64.9	40.2	48.4	**73.6**	23.3	34.4	34.6	**76.3**	47.2
TOUCAN$_{base}$	71.8	74.1	72.9	57.1	72.2	63.7	65.5	46.0	54.1
+Synonyms	72.9	78.4	75.5	64.6	72.2	68.2	70.6	54.0	61.2
+BackTrans	**76.6**	82.1	79.3	62.8	**77.0**	69.1	**75.5**	50.7	60.6
+Syn&BackTrans	74.0	80.6	77.2	67.2	73.8	**70.3**	73.3	55.8	63.3
+Exemplar	75.8	**84.1**	**79.8**	61.0	71.0	65.6	71.2	47.5	56.9
TOUCAN	75.1	80.8	77.9	66.1	74.2	69.9	74.8	56.9	**64.6**

5.4.3 Analysis

5.4.3.1 No Use Categories

Performance on the *No Use* category is lower than on the other categories, so we did some manual investigation to better understand why. Table 5.6 shows some *No Use* examples that were incorrectly labeled by TOUCAN model. We see some clues that seem potentially useful, such as prepositional phrases indicating that the object is not the main focus (e.g., under the window, against the workbench). And "dropped" implies that the object was passive (i.e., something happened to it). But we saw many different types of *No Use* contexts. Focusing on this category could be an interesting direction for future research.

5.4.3.2 Implicit Actions

We also conducted a manual analysis to see how common truly implicit actions are. We randomly sampled 200 examples from the *Used* or *Anticipated Use* sentences in our

Table 5.6: A sample of *No Use* cases that were predicted incorrectly by the system. Our TOUCAN model predicted *Used* for i., ii., and iii., and *Anticipated Use* for iv. and v.

Sentences
i. The couch was crammed under the window with the tv in the corner. ii. Today I dropped my spectacles in the dog kennel again while getting my crazy dog out. iii. She laid out the smock on the wardrobe and moved over to me.
iv. I am now debating taking the cabinet back to Target and exchanging it. v. Duo took a step back and leaned against the workbench.

dataset and judged whether the main predicate explicitly described the action involving the object. This was an informal study, but we judged nearly 30% (58) as having implicit actions. Table 5.7 shows some sentences with implicit actions. We noticed a few common categories and showed their frequencies in Table 5.8.

In 16 sentences, the main verb was underspecified, such as “use”. There were 16 light verb constructions [172], in which the verb has little semantic content of its own. There were 4 metonymic verbs [81, 173] such as “finish” and “start”. In 9 additional cases, the main predicate did not describe the action, but it could be inferred from a prepositional phrase (e.g., sentence 3 in Table 5.7). There are also 13 implicit examples that do not fall into any of these categories.

Table 5.7: The predicate (target) for frame identification is in red. The physical objects are in blue. The red frame represents the action explicitly indicated by the predicate. The blue frame represents the prototypical action associated with the object, which people would infer.

Sentence	Frame Identification	Function of Object
You stand on a street with a guitar and a crowd will come.	Posture	Make_noise
For best results , run the toothpick from the inside out.	Cause_motion	Removing
Wow, that woman on trombone rocked it like nothing I’ve seen or heard before.	Desirability	Make_noise
I braved the heat with my shears and headed for the front yard.	No associated frame	Cutting
When the aids came in and said she had to use the bedpan , she threw a fit.	Using	Excreting
I grabbed my 7x50 binoculars but the coyote has run away.	Manipulation	Perception_experience
She had this spunky , schoolgirl-theme outfit complete with ammo backpack and skirt.[7]	Possession	Wearing

[7] It is worth noting that the *Wearing* frame do contains a lexical unit “have on”. So it is possible to infer a conceptual connection between *Possession* and *Wearing*.

Table 5.8: Manual analysis on how common implicit actions are in a random sample of 200 sentences.

Implicit Type	Count
Underspecified verb	16
Light verb	16
Metonymic verb	4
Prepositional phrase	9
Misc	13

5.5 Related Work

Commonsense knowledge has long been recognized as an essential part of natural language understanding [30, 149, 183]. Some work specifically argued that commonsense knowledge about physical objects is often used to make inferences and plays an important role in narrative text understanding [22].

Recently, a variety of projects have focused on acquiring knowledge about physical objects, including relative physical knowledge [54], relative spatial relations [35], location knowledge [73, 187], and object affordance [122]. Jiang and Riloff [72] (Chapter 4) developed a method to learn the most typical way that people use human-made physical artifacts, and used FrameNet frames as a representation for common object functions. We created a dataset of physical objects annotated with their prototypical functions. Research in this Chapter builds upon that work by developing a model for identifying the usage status of physical objects mentioned in a sentence, which we argue is a necessary precursor to applying prior knowledge about prototypical functions.

Recently, there have been efforts aimed at learning implicit information with pre-trained language models. Weir et al. [178] explored using pre-trained masked language models to capture implicit knowledge elicited from humans, which are so-called stereotypical tacit assumptions. Geva et al. [56] created a question answering dataset consisting of questions that require implicit multi-step reasoning skills, such as *"Did Aristotle Use a Laptop?"*. They show that a large language model fine-tuned on related datasets without retrieval of relevant knowledge performs far worse than humans, and high-quality retrieval makes the model more effective in the reasoning process. Talmor et al. [168] trained language models with automatically generated data sampled from existing knowledge sources. They show that language models can combine implicit knowledge encoded in their parameters with explicit rules and facts, and further perform reasoning.

Our work also has ties to frame semantic parsing [6, 165], which is the task of automatically extracting frame semantic structures from sentences based on the Berkeley FrameNet project [5, 141]. The process begins with identifying frame-evoking words in the sentence (*target identification*) and identifying the evoked frame for each target (*frame identification* [18, 71]). However, one limitation of this setting is that it typically relies on the predicate to predict the frame. For example, the sentence *"Sam enjoyed the book"* would not trigger a reading frame because "enjoy" is not associated with reading in FrameNet. Our work strives to identify this implicit action by recognizing that the book was used and then applying the prototypical function associated with books.

5.6 Conclusion

We introduced a new NLP task, *object use classification*, which identifies whether an object mentioned in a sentence has been used or likely will be used. We introduced a gold standard dataset for this task and showed that all 3 categories (*Used*, *Anticipated Use*, and *No Use*) are common in real sentences. Then we presented a transformer-based architecture for this task that uses two types of data augmentation techniques (synonym/hyponym replacement and back translation) and also exploits exemplar sentences from FrameNet that correspond to an object's prototypical function. The resulting classification model achieves reasonably good performance for this task, although there is room for improvement that we hope will inspire future work on this problem.

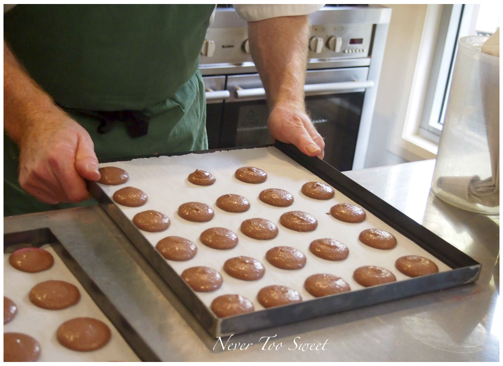
Table 5.7 illustrates the potential for combining our new object use classification model with commonsense knowledge about the prototypical functions of objects in order to improve sentence understanding. Current NLP systems would typically characterize these sentences based on the actions shown in Red, but we argue that the actions shown in Blue are the inferences that humans make when reading these sentences. In future work, we hope to put these pieces together to fully capture both the explicit and implicit meaning behind sentences and the commonsense inferences that people naturally make when reading sentences.

CHAPTER 6

PROTOTYPICAL FUNCTIONS FOR VISUAL ACTIVITY RECOGNITION

In the previous chapters, we have shown that physical objects play an important role in our daily lives. People use different tools to achieve different goals in all kinds of situations. For example, we use a toothbrush to clean our teeth, a microwave oven to heat food, and a camera to take photos. Chapter 4 introduced models that can learn functional knowledge of physical objects. Chapter 5 showed that by using the prior knowledge of functions of physical objects, NLP systems can better understand the implicit meaning of a sentence. In this chapter, we will demonstrate the significance of functional knowledge via a downstream computer vision task.

Physical objects play an important role in computer vision. There are well-established computer vision tasks that aim to identify the objects in an image, such as object detection [92] and object classification [42, 78]. Recently, attention has been paid to more comprehensive image understanding, such as identifying the salient event depicted in an image as well as relevant people and objects. **Situation recognition** [191] is the task of producing a structured summary of an image that describes the main activity and the entities that fill semantic roles for that activity. The task was originally defined using frame structures from FrameNet [5, 141] as the activity representation. For example, given the image shown in Figure 6.1, a system should identify a *baking* event (which is indexed in FrameNet as a type of *Cooking_creation* activity), and recognize the corresponding semantic role/value pairs associated with FrameNet's *Cooking_creation* frame. Models for this task usually follow a two-step pipeline: (1) predict a verb that describes the activity depicted in the image, and (2) identify the entities associated with each semantic role. Previous systems have relied solely on features extracted from the image and have not yet exploited any external commonsense knowledge.



(a) Input image.

BAKING	
ROLE	VALUE
AGENT	MAN
FOOD	COOKIE
FOODCONTAINER	COOKIE SHEET
HEATSOURCE	OVEN
PLACE	KITCHEN

(b) Annotated activity and semantic roles.

Figure 6.1: *Situation Recognition* involves predicting activities with semantic role/value pairs.

For this work, we focus on the activity recognition (verb prediction) part of the situation recognition task. We hypothesize that (a) correctly identifying the activity in an image strongly depends on recognizing the objects that appear in the image, and (b) explicit commonsense knowledge about physical objects can also be beneficial. An intuitive extension to visual reasoning is that if an object appears in an image, especially when it is used by a person, the activity depicted in the image is likely to be the prototypical function associated with the object. For example, a woman holding a comb is probably brushing her hair, and a man holding a cookie sheet (as shown in Figure 6.1) is probably baking.

We explore these hypotheses by creating a transformer-based model that incorporates commonsense knowledge about the prototypical functions of physical objects for visual activity recognition. Our experimental results confirm that correctly identifying the objects in an image is very important for activity recognition, and we show that providing explicit knowledge about the prototypical functions of objects can improve performance for this task.

6.1 Visual Activity Recognition with Object Functions

Given an image, the visual activity recognition task predicts a verb that describes the main activity in the image. Figure 6.2 shows the framework of our model called **ARF** (**A**ctivity **R**ecognition with **F**unctions), which takes 3 sources of input: 1) the image, 2) nouns corresponding to the objects in the image, and 3) the names of FrameNet frames that describe the prototypical functions of the objects. We use the CLIP [134] model, which

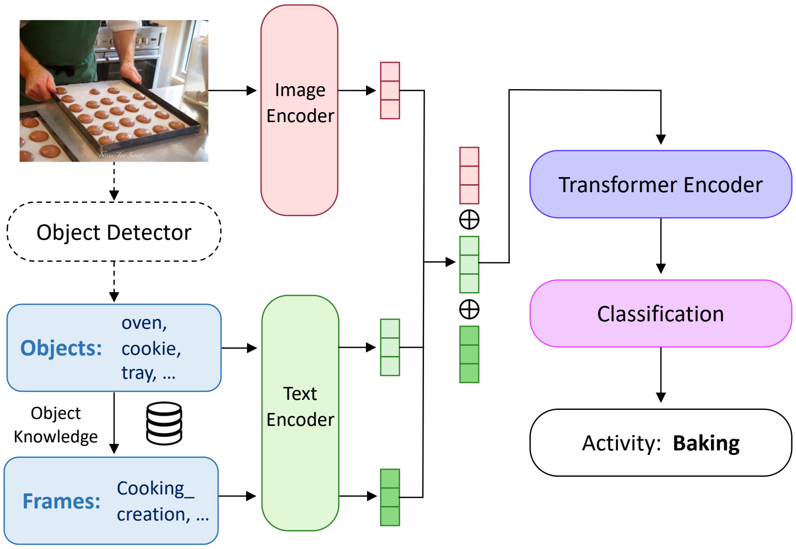


Figure 6.2: Overview of the ARF architecture.

has been pre-trained on both images and text, to generate an encoding for each of the 3 types of input. Finally, we give the concatenated representation vectors as input to a transformer model that is trained to predict a verb for activity recognition.

6.1.1 Notation

The task can be denoted as given the ith image I_i ($i = 1..n$), the system should predict the correct activity verb v_i^*. The score for the jth candidate verb being the activity for image I_i is defined as:

$$\Pr(v_i^j|I_i) = \frac{\exp(g(I_i, v_i^j))}{\sum_{k=1}^{m} \exp(g(I_i, v_i^k))} \tag{6.1}$$

where $g(\cdot)$ is a function produced by our model for scoring the assignment of a verb to the image, and m is the total number of candidate verbs. We use negative log likelihood as our loss function:

$$\mathcal{L} = -\sum_{i=1}^{n} \log \Pr(v_i^*|I_i) \tag{6.2}$$

6.1.2 Object Recognition

Ideally, we would use an Object Detector to identify the objects in an image for our experiments. However, the object detectors that are most readily available use categories that do not cover the range of object types that we need. For example, object detection datasets often contain a number of animate objects such as people and animals. As an alternative, we turned to image captioning systems. For our first set of experiments, we used a state-of-the-art image captioning model called OFA [176] to generate 10 different sentences that describe the image. We set beam size 10 and diversity 10. We then extracted the nouns from these sentences to create a set of words that (hopefully) include the objects.

However, even though the image captioning system often generated reasonable captions, the most relevant objects were frequently omitted from the caption, or misidentified.[1] Since the goal of our research is to determine whether *adding* explicit knowledge about an object improves performance, we cannot truly assess the value of such knowledge when we do not know what objects appear in the image. Developing better methods to identify specific objects in an image is an important direction for future research in computer vision. For now, we continued our investigation by performing additional experiments with the gold nouns in the imSitu dataset. These experiments essentially evaluate the impact of adding object knowledge when the objects have been perfectly identified by an oracle.

6.1.3 Prototypical Function Knowledge

We obtained the knowledge of what an object is typically used for from the dataset we created in Chapter 4 (Jiang and Riloff [72]). The data contains a list of physical objects in the form of WordNet synsets [104], and each object is paired with a human-annotated frame from FrameNet that represents its prototypical function. For example, *knife* is paired with the *Cutting* frame.

For each object in an image, we aim to use its function frame to help with activity identification. However, our prototypical functions and imSitu [191] used different subsets of frames from FrameNet. We felt that it made sense to align them, so we used the inter-frame relations provided by FrameNet to map our prototypical function frames to imSitu's frames.

Each frame relation in the FrameNet data is a directed (asymmetric) relation between two frames. There are 9 different frame relations: *Inheritance, Using, Perspective On, Sub-*

[1] One likely reason is that the images are in low resolution and many objects are small, such as a pencil.

frame, Precedes, Inchoative Of, Causative Of, Metaphor, See Also.[2] Though some frame relations are more important for the purpose of alignment (e.g., *Inheritance* represents an *IS-A* relation), we use all frame relations to maximize the number of matched frames.

For each imSitu frame, we create a mapping to all prototypical function frames that are within one hop via any frame relation. Out of the 161 frames in the imSitu dataset, 74 frames can be mapped to at least one of our function frames. Some unmatched imSitu frames include *Giving*, *Judgment*, and *Discussion*, which are activities that do not require any tools or equipment. But there are also frames that should be matched but are not, e.g., frame "Attack" and "Killing" in the imSitu can not be mapped to "Cause_harm" frame. So there is still room for improvement of a better matching mechanism. Finally, we associate each object with its corresponding imSitu frames.

6.1.4 Activity Recognition Model

We use CLIP ViT-B/32 [134] as the backbone model to encode the image and text. For each example, we first apply CLIP's image encoder to produce an image feature vector. Then we use CLIP's text encoder to generate an embedding for each object (noun) and average the object vectors. For each object, we also collect its prototypical function frames and use CLIP's text encoder again to generate embeddings for each frame's name, then average those vectors. If there is no object, or no associated frame, then we encode an empty string.

Next, we build a transformer model consisting of 6 encoding layers and a classification layer on top. As input, the model takes the concatenation of all 3 vectors (corresponding to image, objects and functions). The classifier then selects the most probable action verb from all 504 candidate verbs used in the imSitu dataset.

6.2 Evaluation

The imSitu data contains 126,102 images, with manually annotated activity verbs and frame structures. We follow the same data split (train 75,702, development 25,200, test 25,200) as Yatskar et al. [191]. We report verb prediction accuracy on both the development and test sets. When fine-tuning the transformer, we use batch size 32, hidden vector

[2] Detailed definitions can be found in Ruppenhofer et al. [141].

dimension 512, AdamW optimizer with learning rate 1e-4 and train for 10 epochs.

6.2.1 Experimental Results

Table 6.1 compares our model with six previous methods described in Section 6.3. The ARF row shows the performance of our basic model using only image input. Our model performs a little better than previous systems, probably due to the CLIP model which is quite good. Also, the other models are trained for the full situation recognition task, whereas our model is trained solely for the verb prediction task.

The next two rows show results when adding embeddings for the nouns extracted from the captioning system (nouns$_C$) and when using the nouns as well as their function frames (nouns$_C$+func). The nouns alone produce just a tiny improvement, but adding the function frames improves a bit more. We believe that these results are primarily due to the limitations of the captioning system.

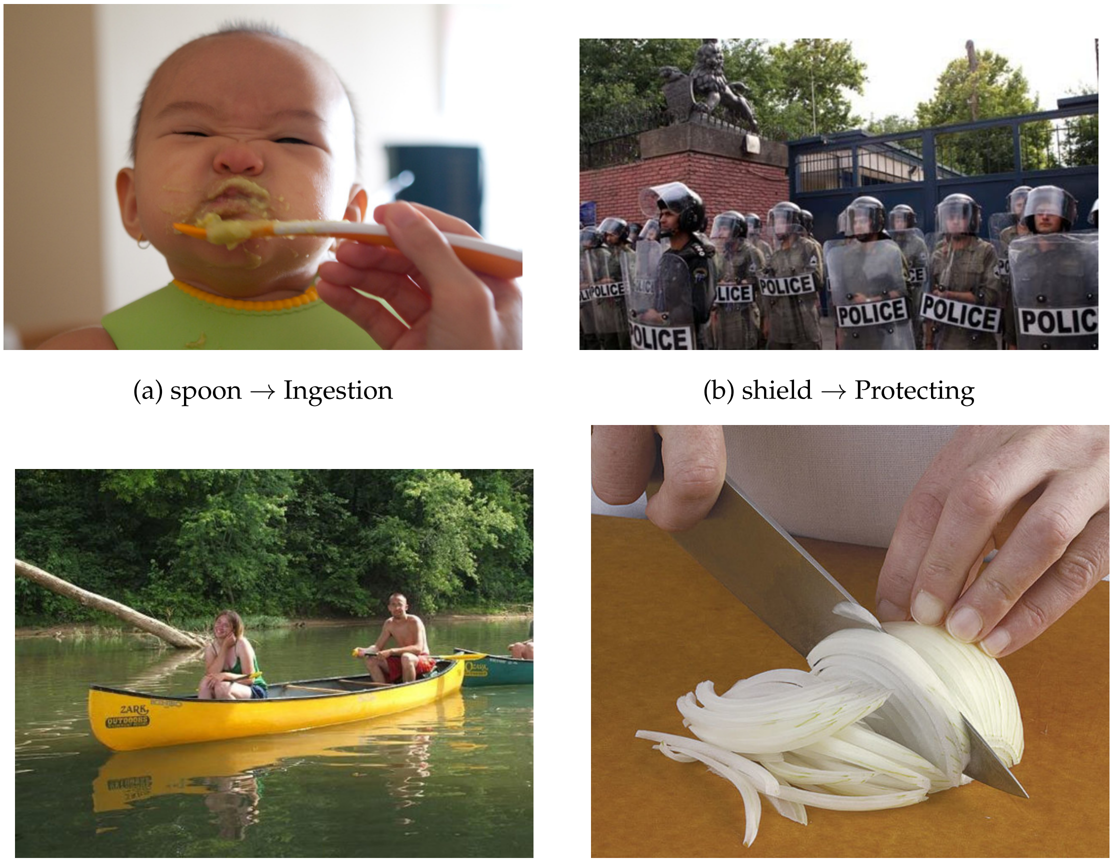
The last two rows in Table 6.1 show the performance when using the gold nouns (nouns$_G$) and when using the gold nouns plus their associated function frames (nouns$_G$+func). These results show a huge performance boost simply from correctly identifying all the objects in the image. And providing the external knowledge about their prototypical functions further improves performance. In the next section, we try to better understand the role that objects play.

6.2.2 Analysis

Figure 6.3 shows some examples of how the functions of objects in the image can help identify the main activity. Consider subfigure (a), we see a hand-held **spoon** in front of the baby's mouth; the baby is expressing their like or dislike by making a grimace; there is some green substance (presumably food) both on the face and spoon. We don't see a series of continuous actions, yet we know it is a feeding event because of our commonsense knowledge. Similarly for the other images in Figure 6.3, from the shields, we can infer *Protecting*; looking at the canoe, we know it is *Motion*; and the knife is a good indicator for *Cutting*.

Table 6.1: Experimental results.

Model	Dev Acc	Test Acc
Yatskar et al. [191]	32.3	32.3
Cooray et al. [36]	38.0	38.2
Pratt et al. [126]	39.6	39.9
Suhail and Sigal [162]	43.2	43.3
Cho et al. [33]	44.4	44.7
Li et al. [86]	-	45.6
ARF	46.2	46.4
ARF+nouns$_C$	46.6	46.5
ARF+nouns$_C$+func	46.9	47.2
ARF+nouns$_G$	69.2	69.5
ARF+nouns$_G$+func	72.0	71.9



(a) spoon → Ingestion

(b) shield → Protecting

(c) canoe → Motion

(d) knife → Cutting

Figure 6.3: Object triggering function frames. The gold verbs are (a) feeding, (b) guarding, (c) floating, and (d) slicing.

6.2.2.1 Images with and without Objects

However, not all images contain “salient” physical objects. For example, imagine a picture showing a man running on a trail. The man is wearing clothes, which usually does not help with identifying the running activity (people generally wear clothes). In order to tease apart the images with and without salient objects, we divided the dev set into two subsets: one set (*w/ Func*) contains 8,957 images where at least one gold noun is associated with a function frame, and the other set (*w/o Func*) contains 16,243 images for which no nouns map to any frames. Since the gold annotations only provide semantic role values that are associated with the main activity, it is safe to assume that the *w/ Func* set of images would contain salient objects. Table 6.2 compares the performance of our systems on each subset of data. We see that performance is nearly identical when only using image features. Adding the gold nouns produces a big performance gain for both groups, although it benefits the *w/ Func* subset a little more. When the function frame knowledge is introduced, we see more separation: the images that depict physical objects associated with functions benefit more from having external knowledge about functions. This result confirms that the knowledge is beneficial in the expected way.

6.2.2.2 Objects with Functions Being Triggered

We plot the objects in the imSitu dataset that are associated with a prototypical function in a word cloud as shown in Figure 6.4, where larger words have a higher frequency in the dataset. We can see that there are a lot of vehicles and commonly used household essentials.

6.2.2.3 Which Semantic Categories Matter?

The performance gap between ARF+nouns$_G$ and ARF is substantial, and we were curious to understand what types of nouns contributed the most. So we conducted another set of experiments on the dev set to identify certain types of semantic roles.

There are 190 different semantic roles in the data, but we are primarily interested in understanding the importance of physical objects. So we coarsely grouped the semantic roles into 3 categories roughly corresponding to *People*, *Locations* and *Objects*. To keep things manageable, we identified the 16 most frequent semantic roles that appear in at least 2,000 images and manually assigned them to the 3 categories. The *People* category

Table 6.2: Comparing performance on images with and without objects that have function frames.

Model	**w/ Func**	**w/o Func**
ARF	46.0	46.4
ARF+nouns$_G$	70.4	68.5
ARF+nouns$_G$+func	75.1	70.4

Figure 6.4: A word cloud of objects that are associated with a prototypical function.

includes *agent*, *agentpart*, *victim*, and *coagent*. The *Locations* category contains *place* and *destination*. The *Objects* category contains *tool*, *item*, *substance*, *object*, *container*, and *vehicle*. We disregarded a few semantic roles that are highly ambiguous (e.g., *source* can be both a location and object).

Table 6.3 shows our experimental results. Each experiment collected all images containing at least one instance of a relevant semantic role and then evaluated performance on those images both with and without the gold annotated nouns. For example, the *Objects* column shows that our model achieved 72.2% accuracy on the images that contain at least one object when it was given the nouns. But performance dropped to 37.2% accuracy on those same images without the nouns. In contrast, providing the gold nouns had much less impact on the other sets of images, which contain *People* or *Locations* but not necessarily *Objects*.

Table 6.3: Performance with and without the nouns for specific semantic roles.

	People	**Locations**	**Objects**
with Nouns	69.3	69.2	72.2
w/o Nouns	61.4	64.4	37.2

6.2.2.4 Salient Objects

Another challenge is how to find the "salient" objects that play important roles in the image, and from which we have a better chance of identifying the main activity. We count the number of physical objects (not in the *People* or *Locations* semantic category) for all images. We find that nearly 40% of images are annotated with two or more objects. In our ARF model, when there are multiple objects in the image, we simply use the average of each object's embedding, which could potentially be improved by giving more weight to the most salient object. This issue may be even more important when using object detection systems because they may identify more objects (the gold annotation only contains objects that belong to a pre-defined semantic role)! This is an important issue to study in future work.

6.3 Related Work

Commonsense knowledge about physical objects has long been recognized to be important for natural language understanding [22]. Within the NLP community, a variety of recent projects have focused on acquiring and using different types of knowledge about physical objects, including relative physical knowledge [54], relative spatial relations [35], semantic plausibility [177] and object affordances [122]. The work most relevant to this research is Jiang and Riloff [72] (Chapter 4), which developed a method to learn the most typical way that people use human-made physical artifacts from text. We used FrameNet frames as a representation for object functions and created a dataset of physical objects paired with their prototypical function frames to evaluate the results. In this chapter, we incorporate the prototypical function data into a transformer-based model for visual activity recognition.

Visual reasoning tasks, such as visual question answering [3] and image captioning [193], have been widely explored for understanding images and videos. Previous work has proposed to use external knowledge for visual tasks, such as image classification [99], object detection [155], and visual question answering [184].

Situation recognition is a task of recognizing the activity depicted in an image, including the people and objects involved in the activity and the roles these participants play. Yatskar et al. [191] introduced the **imSitu** dataset, which associates images with a verb that describes the main action, and a set of semantic roles derived from FrameNet [141]. They tackled this problem by first applying the VGG network [154] to extract features from the image and then building a CRF model to jointly predict the verb and semantic roles. Several research efforts have further explored this task. Suhail and Sigal [162] used a graph neural network to capture the relations between semantic roles. Pratt et al. [126] used a LSTM to jointly classify verbs and semantic roles. Cooray et al. [36] cast situation recognition as a query-based visual reasoning problem and further handled inter-dependencies between queries to overcome the sparsity issues of semantic roles. Recently, Cho et al. [33] proposed a collaborative framework using two transformer modules, and Li et al. [86] used contrastive learning to distinguish the correct activities from negative examples. All of these prior efforts have relied solely on features extracted directly from the image. Our work aims to show that explicitly providing commonsense knowledge about objects can also be beneficial for visual activity recognition.

6.4 Conclusion

The prototypical functions of physical objects is a type of commonsense knowledge that is important for NLP. In this work, we showed that it can be a useful source of information for image understanding as well. Specifically, we tackled the situation recognition task by building a transformer model that incorporates the functions of objects to predict the activity in an image. The experiments show that knowledge of the objects and their prototypical functions can improve performance on this task. However, automatically recognizing the objects in an image remains a challenge, and exploiting better object detection methods is an important direction for future work.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this dissertation, I present research on learning one type of commonsense knowledge — prototypical functions — and demonstrate its importance for artificial intelligence tasks. In this chapter, I will first summarize my research contributions, and then discuss the future directions of this line of work.

7.1 Research Claims and Contributions Revisited

In Chapter 1, I made two research claims and a brief description of my research contributions to support the claims. Here is a revisit of the claims, and I will demonstrate how each chapter relates to them.

> *Claim #1: Commonsense knowledge about the prototypical functions can be learned from large corpora and language models.*

In Chapter 3, I introduced functional knowledge of locations, i.e., people go to different places to engage in activities that reflect their goals. For example, people go to restaurants to eat, libraries to study, and churches to pray. I referred to an activity that represents a common reason *why* people typically go to a location as a *prototypical goal activity*. My research learned functions for specific locations using a text corpus and semi-supervised learning. First, I extracted activities and locations that co-occur in goal-oriented syntactic patterns. Next, I created an *activity profile matrix* and apply a semi-supervised label propagation algorithm to iteratively revise the activity strengths for different locations using a small set of labeled data. I showed that this approach outperforms several baseline methods when judged against goal activities identified by human annotators.

In Chapter 4, I introduced functional knowledge of human-made physical objects, i.e., the typical way how people use the objects, which I referred to as its *prototypical function*. For example, people use a knife for cutting, a car for transportation, and a bed for sleeping. The prototypical function of a physical artifact is a kind of commonsense knowledge that we rely on to understand natural language. I introduced a new NLP task of learning the prototypical uses for human-made physical objects. I selected frames from FrameNet to represent a set of common functions for objects, and described a manually annotated data set of physical objects labeled with their prototypical function. I also presented experimental results for this task, including BERT-based models that use language model predictions from masked patterns as well as artifact sense definitions from WordNet and frame definitions from FrameNet. My model outperforms several baseline methods, including using an existing knowledge base.

> *Claim #2: Commonsense knowledge about the prototypical functions can benefit downstream artificial intelligence applications.*

In Chapter 5, I proposed that commonsense knowledge about the typical functions of physical objects allows people to make inferences during sentence understanding. For example, we infer that *"I enjoyed the book"* means that I enjoyed *reading* the book, even though the action is implicit. In the previous chapters, I have proposed methods to learn the prototypical functions of physical objects in order to enable inferences about implicit actions. But many sentences refer to objects even when they are not used (e.g., *"The book fell"*). I argue that NLP systems need to recognize *whether* an object is being used before inferring *how* the object is used. I defined a new task called *Object Use Classification* that determines whether a physical object mentioned in a sentence was used or likely will be used. I introduced a new dataset with human annotation for this task. The annotation results demonstrate that when a sentence mentions or implies the use of an object, **96%** of the time, it corresponds to the typical function of the object. This shows that object use classification combined with knowledge about the prototypical functions of objects has the potential to yield good inferences about implicit and anticipated actions. I also presented a classification model that exploits data augmentation methods and FrameNet when fine-tuning a pre-trained language model. It substantially outperforms two prompting-based methods with language models.

In Chapter 6, I apply the prior knowledge of physical objects to a computer vision task, *situation recognition*. Situation recognition is the task of recognizing the activity depicted in an image, including the people and objects involved. Previous models for this task typically train a classifier to identify the activity using a backbone image feature extractor. I propose that commonsense knowledge about the objects depicted in an image can also be a valuable source of information for activity identification. I investigate whether this prototypical function knowledge can also be beneficial for visual situation recognition. I build a framework that incorporates this type of commonsense knowledge in a transformer-based model that is trained to predict the action verb for situation recognition. Our experimental results show that adding prototypical function knowledge about physical objects does improve performance for the visual situation recognition task.

7.2 Future Work

7.2.1 Evaluation for Commonsense Knowledge

In Chapter 3, when evaluating system-generated prototypical goal activities against the gold standard, we use the Mean Reciprocal Rank [174] metric to judge the quality of the top activities. Mean Reciprocal Rank is a common evaluation metric for knowledge acquisition tasks, such as knowledge base completion [32] and knowledge graph refinement [118]. However, evaluating the quality of system-generated commonsense knowledge is still a difficult task. As we indicated in Chapter 3 Section 3.2.3, the same meaning can be expressed with many different phrases, and expressions with different meanings can be very similar or highly related. The only way to truly know whether two phrases refer to the same concept is through manual evaluation, which is expensive.

Additionally, this type of evaluation metric depends on a high-quality gold standard dataset. Nowadays, the NLP community favors benchmarks on a large scale. It is a big challenge to build a gold dataset that covers all commonsense knowledge (or even just one type of knowledge) for an intrinsic evaluation of a commonsense knowledge acquisition system. In terms of extrinsic evaluation, previous work proposed various tasks that require commonsense reasoning ability, such as Winograd Schema Challenge [85] and ROCStories [110]. More recently, multiple question answering datasets have been created to evaluate models' commonsense reasoning ability, such as CommonsenseQA [167], ReCoRD [196], OpenBookQA [103], HellaSwag [194], and TruthfulQA [90]. However, instead of focusing on one or a few language processing or reasoning ability, these benchmarks require a more comprehensive mix of language processing and reasoning skills within a single task [160]. For example, in Zhang et al. [196], they categorize the required commonsense knowledge into four coarse categories: conceptual knowledge, causal reasoning, naive psychology and others. Formalizing the commonsense knowledge needed for even simple problems is not easy. For future work, I want to improve both intrinsic and extrinsic evaluation on: 1) how to design better intrinsic evaluation of the automatically acquired pieces of commonsense knowledge and 2) how to create context-based benchmarks that target specific commonsense knowledge, including but not limited to prototypical functions.

7.2.2 Integration of Symbolic and Neural Methods

Neuro-symbolic methods have a long-standing history in the field of artificial intelligence. It still remains a difficult problem as how to combine symbolic and neural models for commonsense knowledge representation, learning and reasoning. A number of previous works tried to integrate symbolic knowledge into neural models. For example, Li et al. [87] learned word embeddings using ConceptNet; Mihaylov et al. [103] builts a knowledge-aware neural model utilizing knowledge from ConceptNet for question answering; ERNIE [163] integrates knowledge into pre-trained language models. It is also an intuitive idea to use neural models to help overcome the coverage issue of commonsense knowledge bases. As we discussed in Chapter 4, COMET [16], a transformer-based framework for automatic construction of commonsense knowledge bases, does help improve the coverage of ConceptNet since it was trained on ATOMIC and ConceptNet by adapting the weights of language models to learn to produce novel and diverse commonsense knowledge tuples. These works inspired us to consider combining functional knowledge (symbolic form) with neural frameworks for a better representation and utilization. In Chapter 4, when using frames as a representation for prototypical functions, we simply encode the frames using their names and definitions with language models. It is worth investigating techniques other than contextual embeddings directly from pre-trained language models and taking advantage of the rich structure of frames to form a better neural representation of frames. This also applies to the methods in Chapter 6, in which we try to incorporate the knowledge of objects into the neural networks.

7.2.3 Implicit Language Understanding

Understanding the implicit information from text remains a big challenge in NLP. Implicitness is common in language. In Chapter 5, we show that when a sentence implies a use of the mentioned object, 30% of the time the main action is implicit (e.g., light verbs [172], metonymic verbs [173], etc.). Implicitness could also exist in different levels, such as metaphor [153], simile [130] and enthymeme [136]. A lot of key questions still remain unexplored. For example, little work has been done to understand how well existing textual inference datasets capture implicitness and how well current models understand implicit language. An interesting topic will be whether we can extract testing examples targeting implicit language understanding from existing benchmarks for different NLP tasks, such as textual entailment and question answering. Recent advances in large language models provide efficient ways to capture implicit semantic information from unsupervised training. As Bommasani et al. [14] propose, the rise of language models trained on broad data initiated the foundation model paradigm in NLP. Language models rapidly became the foundations for almost all modeling work [88]. A critical question has attracted attention in the community as to what extent language models understand commonsense knowledge and use them for implicit information understanding.

7.2.4 Frame Semantics

Some previous work on commonsense knowledge acquisition has opted to collect natural language words and phrases as a representation [138, 159]. One disadvantage of using natural phrases is that there exist different ways to express the same meaning, making the representations redundant and difficult to evaluate. As discussed in Chapter 4, we selected frames as a canonical representation for the knowledge of prototypical functions. However, there also exist shortcomings with this representation. Our annotation results indicate that not every function can be mapped to an appropriate corresponding frame in FrameNet. So an interesting avenue for future research is to expand FrameNet frames in order to achieve better coverage for other domains. Besides, as we briefly mentioned in Chapter 1, current FrameNet annotations focus more on the explicit meaning of the sentence, i.e., it usually relies on the explicit predicate to infer the meaning of the sentence. In Chapter 5 Section 5.4.3, we also show that sometimes knowledge of mentioned objects plays a more important role in sentence understanding. As a result, I am interested in expanding the FrameNet dataset so that it can also test the inference ability of NLP systems.

7.2.5 Commonsense Knowledge for Multimodality

Multimodal tasks such as visual question answering and image captioning have attracted a considerable amount of interest in both Computer Vision (CV) and NLP communities. However, not many existing crossmodal architectures are focusing on commonsense knowledge [186]. It is worth investigating how to inject commonsense knowledge into a multimodal model through textual, or visual input. Chapter 6 shows some results that identifying what objects exist in the image and understanding their prototypical functions can help identify the correct action depicted in the image. We demonstrate that there is still substantial room for improvement of the state-of-the-art object detection systems compared to using the gold annotations. I would argue that further efforts need to be made to 1) explore unifying the vector representation space between textual and visual input for multimodal tasks and 2) reconsider what types of commonsense knowledge NLP systems share with CV systems and what types of knowledge they should include distinctively.

APPENDIX A

LIST OF MERGED ACTIVITIES

{ER {}, ERROR ERROR, ERROR {}}

{ER {}, ERROR ERROR}

{ER {}, ERROR {}}

{ERROR ERROR, ERROR {}}

{achieve relaxation, feel relaxation, get relaxation, relax {}}

{achieve relaxation, relax {}}

{acquire degrees, earn degrees, get degrees}

{acquire goods, get items, obtain goods}

{acquire help, get help}

{adjust backs, adjust spine, fix backs}

{admire flora, see flowers, view flowers}

{admire flowers, view flowers}

{admire history, see history}

{admire ocean, see ocean}

{admire scenery, admire view, see view}

{adopt animals, adopt pet, adopt pets}

{adopt animals, adopt pets}

{adopt child, adopt children}

{adopt dog, adopt dogs, adopt puppy}

{adopt pet, adopt pets}

{alleviate pain, fix pain}

{argue case, argue cases}

{arrive destination, reach destinations}

{ask for date, find dates, get dates}

{atone crimes, atone {}}

{barbecue {}, grill meat}

{barbeque {}, grill meat}

{be entertained, watch entertainment}

{be outdoors, experience outdoors}

{be punished, receive punishment}

{become educated, gain education, gain knowledge, get educated, learn {}, receive education}

{become educated, gain knowledge, get educations, learn {}, receive education}

{become famous, become stars, get famous}

{board boats, board ship, board ships}

{bowl ball, bowl balls, bowl {}, go bowling, play bowling}

{brainstorm ideas, communicate ideas, voice ideas}

{browse links, browse {}}

{browse sites, browse websites, read websites}

{browse websites, view websites}

{bury bodies, bury body}

{buy alcohol, buy booze, buy liquor}

{buy bicycles, buy bikes, purchase bicycles, purchase bikes, purchases bicycles {}}

{buy books, purchase books}

{buy car, buy cars}

{buy clothes, purchase clothes, purchase clothing}

{buy cosmetics, buy make-up, buy makeup, purchase makeup}

{buy dress, buy dresses}

{buy drugs, buy medicine}

{buy fabric, buy textiles, purchase cloth}

{buy flowers, purchase bouquets, purchase flowers}

{buy food, buy groceries, purchase food}

{buy food, buy groceries, purchase groceries}

{buy food, buy groceries}

{buy foodstuffs, buy groceries}

{buy furniture, purchase furniture}

{buy games, buy video games, purchase games}

{buy gas, buy gasoline}

{buy goods, buy items, buy merchandise, buy sundries}

{buy goods, buy items, buy products}

{buy goods, buy items, purchase items}

{buy goods, buy things}

{buy goods, shop {}}

{buy goods, shop items}

{buy iPhone, buy iPhones, buy iphones}

{buy materials, buy supplies}

{buy medicine, buy pharmaceuticals}

{buy office supplies, buy supplies}

{buy records, purchase albums, purchase vinyl}

{buy sandwich, buy sandwiches}

{buy supplements, purchase supplements}

{buy vitamins, purchase vitamins}

{camp out, camp {}}

{catch bus, catch buses, catch rides, ride bus, ride buses}

{catch fish, fish {}, go fishing}

{catch flight, catch flights}

{catch ride, catch rides, get rides, take transportation}

{catch waves, enjoy waves}

{celebrate events, celebrate occasion}

{checkout book, checkout books}

{choose foods, choose meal, decide food, order food}

{climb up, climb {}}

{complete work, finish work}

{dance {}, go dance}

{discard trash, dispose trash, dispose waste, dump garbage, dump trash, empty garbage, toss garbage}

{discover treasure, find treasure}

{dispose trash, toss trash}

{do studying, study {}}

{do time, serve sentences, serve time}

{do work, perform work, work {}}

{donate goods, donate {}, make donation}

{drink beer, drink beers}

{drink beer, drink lager}

{drink beverages, drink drinks}

{drive around, drive {}}

{drive boat, pilot boats}

{drive places, drive somewhere}

{earn degree, gain degrees, get degree, get degrees}

{earn degree, get degree}

{earn degrees, get degree}

{earn money, make money}

{earn paycheck, make money}

{eat food, eat meal, have meals}

{eat food, eat meals}

{eat sandwich, eat sandwiches}

{empty bladders, pee {}}

{enjoy relaxation, get relaxed}

{enjoy sun, enjoy sunshine}

{fight battles, fight wars}

{find boots, select boots}

{find crafts, get crafts}

{find cup, get cup, get mug}

{find deals, get deals}

{find employment, get employed, get job, get jobs}

{find facts, find truth}

{find flowers, get flowers, order flowers}

{find information, gather data, search information, seek information}

{find information, gather information, get information, research information, research {}}

{find information, gather information, search information}

{find information, get information, learn information, read information}

{find information, seek information}

{find medicine, get medicine, get prescriptions}

{find vitamins, get vitamins}

{fix ailments, fix diseases, treat diseases}

{fix bodies, heal body, repair body}

{fix car, fix cars, repair car}

{fix roofs, repair shingles}

{fix shingles, repair roofing, repair shingle, repair shingles}

{further education, gain education, gain knowledge, receive education}

{gain education, gain knowledge, get educated, receive education}

{gain education, get education, learn information, receive education}

{gain education, learn {}, receive education}

{get bargains, get deals}

{get drugs, get medication, pick up medication, pickup medicine}

{get food, obtain food, retrieve food}

{get haircut, get haircuts, receive haircut}

{get help, seek assistance, seek help}

{get information, learn information, view information}

{get information, obtain information}

{get mail, pickup mail, retrieve mail}

{get massage, get massages}

{get medications, get prescriptions, pick up medicine}

{get medicine, get prescription, obtain medicine}

{get medicine, get prescription}

{get medicine, obtain medicines}

{get parts, take parts}

{get passports, receive passport}

{get paychecks, receive paycheck}

{get permits, obtain permits}

{get plates, grab dishes, retrieve dishes}

{get plates, grab dishes}

{get raises, negotiate raise, request raises}

{get rest, rest {}}

{get services, obtain services}

{get shelter, seek shelter}

{get treated, seek medical treatment}

{get treatment, receive medical treatment}

{get treatment, receive therapy}

{get treatment, receive treatment, seek treatment}

{get warm, warm body}

{go shopping, shop {}}

{go sledding, ride sled}

{go swimming, swim {}}

{have vacation, take vacation}

{help animals, protect animals, save animals}

{help baby, help infants, treat baby}

{hike {}, take hikes}

{increase knowledge, learn facts, learn information}

{launch boat, launch boats}

{lay down, lie down}

{learn information, learn knowledge}

{learn law, study law}

{mail letter, mail letters, send letters}

{mail packages, send package, send packages}

{make call, make calls}

{meet agents, see agent}

{meet ambassadors, see ambassador}

{meet diplomats, see diplomat}

{meet stars, see celebrities, see stars, visit celebrities}

{mow grass, mow lawn}

{obtain tasks, receive assignments}

{open door, open doors}

{park car, park cars}

{pledge allegiance, show allegiance}

{pray {}, say prayers}

{pray prayers, pray {}, say prayer, say prayers}

{push boundaries, push limits, test limits}

{read books, study books}

{receive aid, receive help}

{relieve selves, relieve themselves}

{rescue animals, rescue pets, save animals}

{resolve conflicts, settle conflict}

{see Mickey, see mickey}

{see art, view art}

{see family, visit family, visit relatives}

{see family, visit family}

{see film, see films, see movie, watch cinema, watch film, watch films, watch movies}

{see film, see films, see movies, watch films, watch movies}

{see friends, visit friends}

{see performance, see performances, see productions}

{see photos, see pictures}

{see videos, view videos, watch videos}

{serve sentence, serve sentences, serve time}

{sleep soundly, sleep {}}

{travel distances, travel {}}

{travel places, travel {}}

{view stars, watch stars}

{wait there, wait {}}

{watch TV, watch television, watch tv}

{watch film, watch films, watch movie, watch movies}

{watch game, watch games}

{withdraw money, withdrawal money}

{worship God, worship {}}

{worship gods, worship {}}

APPENDIX B

PROTOTYPICAL GOAL ACTIVITIES ANNOTATION GUIDELINES

For this task, we will give you a list of locations, which may be specific places (e.g., Disneyland) or general types of places (e.g., beach). Some examples may not be actual locations, but will act as locations in the phrase "go to X". For example, "go to doctor" is an expression that really means "go to the doctor's office". So consider each case as if it appeared in the phrase "go to X".

For each location, think about WHY a person would go there. What activities would someone go there to do? PLEASE LIST THE PRIMARY ACTIVITIES THAT YOU WOULD TYPICALLY EXPECT A PERSON TO DO AT THE LOCATION. Please do NOT list activities that may be incidental (e.g., walking occurs in stores but people generally do not go to a store for the purpose of walking), or activities that could occur at almost any location (e.g., breathing).

Please list up to 3 typical activities for each location. List them in order, so that the first activity in the most typical, the second activity is the second-most typical, etc. (Don't overthink this, just go with your gut instinct.) Try to list 3 activities if you can, but if a location has just 1 or 2 highly typical activities in your mind, then it is ok to list fewer than 3. Also, do NOT list synonyms as different activities! Each activity that you list should be distinct from the others.

Then please list their synonyms, as many as possible. For example, for buy clothing, you can list "buy clothes", "purchase clothes", "purchase clothing". Or for attend meeting, you can list "attend meeting = have meeting". But please be conservative. We hope your answers are accurate.

Please do not use a search engine to look for common activities at the location! We want to know what activities are strongly associated with each location in YOUR mind.

Each activity should be of the form "VERB" or "VERB NOUN". Please use the ROOT form of the verb (i.e., you should be able to put "to" in front of the verb to create an infinitive form). For example, "visit person" (as in "to visit" someone) is preferred over "visiting person" or "visited person". Please try to list fairly general types of activities, although you can be specific if you feel that a location is associated with a very specific type of activity.

Here are a few examples to illustrate the intended task:

```
LOCATION = desert
Activity 1:  hike
Activity 2:  camp
Activity 3:  see wildflowers

LOCATION = Snowbird
Activity 1:  ski
Activity 2:  snowboard
Activity 3:  ride tram

LOCATION = prison
Activity 1:  serve time
Activity 2:  visit person
Activity 3:  -
```

Note that a NOUN is included with the VERB when the VERB alone would be too general (e.g., "see" is not very specific to deserts!).

APPENDIX C

PROTOTYPICAL FUNCTIONS OF OBJECTS ANNOTATION GUIDELINES

The task is to identify the most typical uses for entities, i.e. what does it help a person accomplish? If someone buys or acquires an object, what is your best guess as to what they intend to do with it? If someone goes to a location entity, what is your best guess as to what they intend to do there?

Select one activity (red) that best describes the typical use. Choose "None" if no option fits.

Select one entity category (blue) that the entity belongs to. Choose "None" if no option fits.

You should read the definition of each option to understand what activity it represents (do not judge based on its name). We also show lexical units (LU) that could represent the activity. Click "See More" to display full definition and full list of lexical units.

Once submitted, your answers can NOT be changed anymore. And please do NOT click "Return" button, it submits empty answer.

Examples

Entity	**Activity**	**Category**	**Explanation**
puppet	Performing_arts	None	Performance is a typical use for a puppet.
washing machine	Removing (cleaning)	None	Washing (represented by "Removing" frame) is a typical use for a washing machine.
oil painting	None	Physical_artworks	Oil painting is a type of physical artwork.
tie	Wearing	Clothing	Wearing is a typical use for a tie. It is also a clothing.

APPENDIX D

CONTEXTUAL OBJECT USAGE STATUS ANNOTATION GUIDELINES

Given a sentence that mentions a physical object, your task is to read the sentence and determine if you would assume that the object has been or will be used. For each sentence, please choose one of the options below.

X. “Mistake” – The example has an error because the given definition of the object word is different from its meaning in the sentence. For example, suppose the object “yoyo” is defined as “a toy consisting of a circular object that can be made to go up and down a long piece of string to which it is tied”. Suppose you are given the sentence:

```
Investment yoyos are high-risk but high-reward financial
speculations.
```

You should select X. because “yoyos” does not refer to the toys in the sentence.

A. “Used” – The sentence either describes 1) an action in which the object is/was being used (by the author or someone else), or 2) an action that directly resulted from the use of the object. For example,

```
I played with my yoyo.
I improved my yoyo skills over the last year.
She loves competing with her yoyo.
He broke the window with his yoyo.
He killed the snake with his yoyo.
```

B. “Anticipated Use” – The sentence either states that 1) the object will be used in the future, or 2) implies that someone will presumably use the object. For example,

```
I traded my lunch money for a yoyo.

I went to my room and got my yoyo.

He went to Utah to compete at the state fair with his yoyo.
```

C. "No Use" – Not A or B. For example,

```
The yoyo fell from the shelf.

His yoyo is red.

She lost her yoyo.

The yoyo is a very popular toy.
```

REFERENCES

[1] R. B. Aharon, I. Szpektor, and I. Dagan, “Generating entailment rules from framenet,” in *Proceedings of the ACL 2010 Conference Short Papers*, 2010, pp. 241–246.

[2] Y. S. Alam, “Decision trees for sense disambiguation of prepositions: Case of over,” in *Proceedings of the Computational Lexical Semantics Workshop at HLT-NAACL 2004*, 2004.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015)*, 2015.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “Dbpedia: A nucleus for a web of open data,” in *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC 2007 ASWC 2007)*. Springer, 2007.

[5] C. F. Baker, C. J. Fillmore, and J. B. Lowe, “The Berkeley FrameNet project,” in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING 1998)*, 1998.

[6] C. F. Baker, M. Ellsworth, and K. Erk, “SemEval-2007 task 19: Frame semantic structure extraction,” in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, 2007.

[7] ——, “Semeval-2007 task 19: Frame semantic structure extraction,” in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, 2007.

[8] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, “Abstract meaning representation for sembanking,” in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[9] P. Baumgartner and A. Burchardt, “Logic programming infrastructure for inferences on framenet,” in *European Workshop on Logics in Artificial Intelligence*. Springer, 2004, pp. 591–603.

[10] D. R. Beddiar, M. S. Jahan, and M. Oussalah, “Data expansion using back translation and paraphrasing for hate speech detection,” *Online Social Networks and Media*, vol. 24, p. 100153, 2021.

[11] M. Berland and E. Charniak, “Finding parts in very large corpora,” in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999)*, 1999.

[12] T. Blevins and L. Zettlemoyer, “Moving down the long tail of word sense disambiguation with gloss informed bi-encoders,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, 2020.

[13] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250.

[14] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.

[15] T. Bosc and P. Vincent, “Auto-encoding dictionary definitions into consistent word embeddings,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, 2018.

[16] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, “COMET: Commonsense transformers for automatic knowledge graph construction,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, 2019.

[17] A. Bosselut, R. Le Bras, and Y. Choi, “Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering,” in *Proceedings of the AAAI conference on Artificial Intelligence (AAAI 2021)*, 2021.

[18] T. Botschen, I. Gurevych, J.-C. Klie, H. Mousselly-Sergieh, and S. Roth, “Multimodal frame identification with multilingual evaluation,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, 2018.

[19] G. H. Bower, “Plans and goals in understanding episodes,” *Advances in Psychology*, vol. 8, pp. 2–15, 1982.

[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” in *Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[21] M. Brysbaert, A. B. Warriner, and V. Kuperman, “Concreteness ratings for 40 thousand generally known english word lemmas,” *Behavior Research Methods*, vol. 46, no. 3, pp. 904–911, 2014.

[22] M. H. Burstein, “The use of object-specific knowledge in natural language processing,” in *Proceeding of the 17th Annual Meeting on Association for Computational Linguistics (ACL 1979)*, 1979.

[23] K. Burton, N. Kasch, and I. Soboroff, “The icwsm 2011 spinn3r dataset,” in *Proceedings of the Fifth Annual Conference on Weblogs and Social Media (ICWSM-2011)*, 2011.

[24] K. Burton, A. Java, I. Soboroff *et al.*, “The icwsm 2009 spinn3r dataset,” in *Third*

Annual Conference on Weblogs and Social Media (ICWSM 2009), 2009.

[25] J. G. Carbonell, "Subjective understanding: Computer models of belief systems," Ph.D. dissertation, Yale University, 1979.

[26] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2008)*, 2008.

[27] ——, "Unsupervised learning of narrative schemas and their participants," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-AFNLP 2009)*, 2009.

[28] A. Chang, M. Savva, and C. D. Manning, "Learning spatial knowledge for text to 3D scene generation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

[29] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng, "Mining semantic affordances of visual object categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, 2015, pp. 4259–4267.

[30] E. Charniak, "Toward a model of children's story comprehension," Ph.D. dissertation, MIT, 1972.

[31] S. Chaturvedi, D. Goldwasser, and H. Daumé III, "Ask, and shall you receive? understanding desire fulfillment in natural language text," in *Processings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*, 2016.

[32] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan, "Knowledge graph completion: A review," *Ieee Access*, vol. 8, pp. 192 435–192 456, 2020.

[33] J. Cho, Y. Yoon, and S. Kwak, "Collaborative transformers for grounded situation recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, 2022.

[34] A. G. Cohn and J. Renz, "Qualitative spatial representation and reasoning," *Foundations of Artificial Intelligence*, vol. 3, pp. 551–596, 2008.

[35] G. Collell, L. Van Gool, and M.-F. Moens, "Acquiring common sense spatial knowledge through implicit spatial templates," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*, 2018.

[36] T. Cooray, N.-M. Cheung, and W. Lu, "Attention-based context aware reasoning for situation recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, 2020.

[37] R. E. Cullingford, "Script application: Computer understanding of newspaper stories," Ph.D. dissertation, Yale University, 1978.

[38] D. Das, N. Schneider, D. Chen, and N. A. Smith, "Probabilistic frame-semantic

parsing," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2010)*, 2010.

[39] D. Das, D. Chen, A. F. Martins, N. Schneider, and N. A. Smith, "Frame-semantic parsing," *Computational Linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[40] E. Davis and G. Marcus, "Commonsense reasoning and commonsense knowledge in artificial intelligence," *Communications of the ACM*, vol. 58, no. 9, pp. 92–103, 2015.

[41] J. Davison, J. Feldman, and A. Rush, "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 2019)*, 2019.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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 (NAACL 2019)*, 2019.

[44] H. Ding and E. Riloff, "Acquiring knowledge of affective events from blogs using label propagation," in *Processings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*, 2016.

[45] J. Dunietz, L. Levin, and J. Carbonell, "The effects of lexical resource quality on preference violation detection," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013.

[46] D. Elson and K. McKeown, "Building a bank of semantically encoded narratives," in *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC-2010)*, 2010.

[47] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial Intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[48] O. Etzioni, A. Fader, J. Christensen, S. Soderland *et al.*, "Open information extraction: The second generation," in *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer, 2011.

[49] S. Feng, J. S. Kang, P. Kuznetsova, and Y. Choi, "Connotation lexicon: A dash of sentiment beneath the surface meaning," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-2013)*, 2013.

[50] Ó. Ferrández, M. Ellsworth, R. Muñoz, and C. F. Baker, "Aligning FrameNet and WordNet based on semantic neighborhoods," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 10)*, 2010.

[51] C. J. Fillmore, “Frame semantics and the nature of language,” in *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, vol. 280 (1), 1976, pp. 20–32.

[52] ——, “Frame semantics,” in *Linguistics in the Morning Calm, The Linguistic Society of Korea (ed.)*. Seoul: Hanshin Publishing Company., 1982, pp. 111–137.

[53] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, “Background to framenet,” *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[54] M. Forbes and Y. Choi, “Verb physics: Relative physical knowledge of actions and objects,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, 2017.

[55] Q. Gao, S. Yang, J. Chai, and L. Vanderwende, “What action causes this? Towards naive physical action-effect prediction,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018.

[56] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, “Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 2021.

[57] R. Girju, A. Badulescu, and D. Moldovan, “Automatic discovery of part-whole relations,” *Computational Linguistics*, vol. 32, no. 1, pp. 83–135, 2006.

[58] A. B. Goldberg, N. Fillmore, D. Andrzejewski, Z. Xu, B. Gibson, and X. Zhu, “May all your wishes come true: A study of wishes and how to recognize them,” in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2009)*, 2009.

[59] A. Goyal, E. Riloff, and H. Daumé III, “Automatically producing plot unit representations for narrative text,” in *Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing (EMNLP-2010)*, 2010.

[60] ——, “A computational model for plot units,” *Computational Intelligence*, vol. 29, no. 3, pp. 466–488, 2013.

[61] M. Granroth-Wilding and S. Clark, “What happens next? Event prediction using a compositional neural network model,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2016)*, 2016.

[62] C. Green, “Application of theorem proving to problem solving,” in *Readings in Artificial Intelligence*. Elsevier, 1981, pp. 202–222.

[63] S. Hartmann, I. Kuznetsov, T. Martin, and I. Gurevych, “Out-of-domain FrameNet semantic role labeling,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, 2017.

[64] P. J. Hayes, “The naive physics manifesto,” in *Expert Systems in the Microelectronic Age*. Edinburgh University Press, 1979.

[65] K. M. Hermann, D. Das, J. Weston, and K. Ganchev, "Semantic frame identification with distributed word representations," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, 2014.

[66] J. R. Hobbs, W. Croft, T. Davies, D. Edwards, and K. Laws, "Commonsense metaphysics and lexical semantics," *Computational Linguistics*, vol. 13, pp. 241–250, 1987.

[67] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[68] L. Huang, C. Sun, X. Qiu, and X. Huang, "GlossBERT: BERT for word sense disambiguation with gloss knowledge," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, 2019.

[69] B. Jans, S. Bethard, I. Vulić, and M. F. Moens, "Skip n-grams and ranking functions for predicting script events," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, 2012.

[70] S. Jastrzębski, D. Bahdanau, S. Hosseini, M. Noukhovitch, Y. Bengio, and J. Cheung, "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*, 2018.

[71] T. Jiang and E. Riloff, "Exploiting definitions for frame identification," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, 2021.

[72] ——, "Learning prototypical functions for physical artifacts," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, 2021.

[73] ——, "Learning prototypical goal activities for locations," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018.

[74] R. Johansson and P. Nugues, "LTH: Semantic structure extraction using nonprojective dependency trees," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, 2007.

[75] S. Kalkan, N. Dag, O. Yürüten, A. M. Borghi, and E. Şahin, "Verb concepts from affordances," *Interaction Studies*, vol. 15, no. 1, pp. 1–37, 2014.

[76] G. Kazeminejad, C. Bonial, S. W. Brown, and M. Palmer, "Automatically extracting qualia relations for the rich event ontology," in *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, 2018.

[77] O. Kolomiyets, P. Kordjamshidi, M.-F. Moens, and S. Bethard, "SemEval-2013 task 3: Spatial role labeling," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

[78] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[79] S. Kumar, S. Jat, K. Saxena, and P. Talukdar, "Zero-shot word sense disambiguation using sense definition embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, 2019.

[80] S. Lahiri, "Complexity of word collocation networks: A preliminary structural analysis," in *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, 2014.

[81] M. Lapata and A. Lascarides, "A probabilistic account of logical metonymy," *Computational Linguistics*, vol. 29, no. 2, pp. 261–315, 2003.

[82] W. G. Lehnert, "Plot units and narrative summarization," *Cognitive Science*, vol. 5, no. 4, pp. 293–331, 1981.

[83] W. G. Lehnert and M. H. Burstein, "The role of object primitives in natural language processing," in *Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI 1979)*, 1979.

[84] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[85] H. Levesque, E. Davis, and L. Morgenstern, "The winograd schema challenge," in *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[86] M. Li, R. Xu, S. Wang, L. Zhou, X. Lin, C. Zhu, M. Zeng, H. Ji, and S.-F. Chang, "Clip-event: Connecting text and images with event structures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, 2022.

[87] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Commonsense knowledge base completion," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 2016.

[88] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar *et al.*, "Holistic evaluation of language models," *arXiv preprint arXiv:2211.09110*, 2022.

[89] M. Light and W. Greiff, "Statistical models for the induction and use of selectional preferences," *Cognitive Science*, vol. 26, no. 3, pp. 269–281, 2002.

[90] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

[91] T. Lin, O. Etzioni *et al.*, "Identifying functional relations in web text," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, 2010.

[92] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L.

Zitnick, “Microsoft coco: Common objects in context,” in *Proceedings of the European Conference on Computer Vision (ECCV 2014)*, 2014.

[93] H. Liu and P. Singh, “Conceptnet—a practical commonsense reasoning tool-kit,” *BT Technology Journal*, vol. 22, no. 4, pp. 211–226, 2004.

[94] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[95] S. Longpre, Y. Lu, Z. Tu, and C. DuBois, “An exploration of data augmentation and sampling techniques for domain-agnostic question answering,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019.

[96] S. L. Lytinen, “Conceptual dependency and its descendants,” *Computers & Mathematics with Applications*, vol. 23, no. 2-5, pp. 51–73, 1992.

[97] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014) System Demonstrations*, 2014.

[98] I. Manotas, N. P. A. Vo, and V. Sheinin, “LiMiT: The literal motion in text dataset,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[99] K. Marino, R. Salakhutdinov, and A. Gupta, “The more you know: Using knowledge graphs for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[100] J. McCarthy, “Programs with common sense,” in *Mechanisation of Thought Processes*. Her Majesty’s Stationery Office, 1959.

[101] S. McGregor and E. Ježek, “A distributional model of affordances in semantic type coercion,” in *Proceedings of the 13th International Conference on Computational Semantics (IWCS 2019)*, 2019.

[102] N. McIntyre and M. Lapata, “Learning to tell tales: A data-driven approach to story generation,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-AFNLP 2009)*, 2009.

[103] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, “Can a suit of armor conduct electricity? a new dataset for open book question answering,” *arXiv preprint arXiv:1809.02789*, 2018.

[104] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[105] M. Minsky, “A framework for representing knowledge,” in *P. Winston (Ed.), The Psychology of Computer Vision*. McGraw-Hill, 1975.

[106] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. Berg, T. Berg, and H. Daumé III, "Midge: Generating image descriptions from computer vision detections," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, 2012.

[107] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, "Never-ending learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 15)*, 2015.

[108] A. Modi, I. Titov, V. Demberg, A. Sayeed, and M. Pinkal, "Modeling semantic expectation: Using script knowledge for referent prediction," *Transactions of the Association for Computational Linguistics*, vol. 5, 2017.

[109] R. C. Moore, *The role of logic in knowledge representation and commonsense reasoning*. SRI International. Artificial Intelligence Center, 1982.

[110] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, 2016.

[111] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, 2001, pp. 2–9.

[112] OpenAI, "Chatgpt," 2023.

[113] ——, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[114] M. Palmer, "Semlink: Linking propbank, verbnet and framenet," in *Proceedings of the Generative Lexicon Conference*, 2009.

[115] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," *Computational linguistics*, vol. 31, no. 1, pp. 71–106, 2005.

[116] A. Pancholy, M. R. L. Petruck, and S. Swayamdipta, "Sister help: Data augmentation for frame-semantic role labeling," in *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, 2021.

[117] C. L. Paris, "Tailoring object descriptions to a user's level of expertise," *Computational Linguistics*, vol. 14, no. 3, pp. 64–78, 1988.

[118] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, 2017.

[119] E. Pavlick, T. Wolfe, P. Rastogi, C. Callison-Burch, M. Dredze, and B. Van Durme, "FrameNet+: Fast paraphrastic tripling of FrameNet," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International*

Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), 2015.

[120] H. Peng, S. Thomson, S. Swayamdipta, and N. A. Smith, “Learning joint semantic parsers from disjoint data,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, 2018.

[121] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP-2014)*, 2014.

[122] M. Persiani and T. Hellström, “Unsupervised inference of object affordance from text corpora,” in *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 2019.

[123] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, “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 2019)*, 2019.

[124] K. Pichotta and R. Mooney, “Statistical script learning with multi-argument events,” in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, 2014.

[125] K. Pichotta and R. J. Mooney, “Learning statistical scripts with lstm recurrent neural networks,” in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*, 2016.

[126] S. Pratt, M. Yatskar, L. Weihs, A. Farhadi, and A. Kembhavi, “Grounded situation recognition,” in *Proceedings of the European Conference on Computer Vision (ECCV 2020)*, 2020.

[127] J. Pustejovsky, “The generative lexicon,” *Computational Linguistics*, vol. 17, no. 4, pp. 409–441, 1991.

[128] J. Pustejovsky, C. Havasi, J. Littman, A. Rumshisky, and M. Verhagen, “Towards a generative lexical resource: The brandeis semantic ontology,” in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 06)*, 2006.

[129] J. Pustejovsky, P. Kordjamshidi, M.-F. Moens, A. Levine, S. Dworman, and Z. Yocum, “SemEval-2015 task 8: SpaceEval,” in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

[130] A. Qadir, E. Riloff, and M. A. Walker, “Automatically inferring implicit properties in similes,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, 2016.

[131] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, “Is chatgpt a general-purpose natural language processing task solver?” *arXiv preprint arXiv:2302.06476*, 2023.

[132] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," *OpenAI*, 2018.

[133] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[134] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML 2021)*, 2021.

[135] E. Rahimtoroghi, J. Wu, R. Wang, P. Anand, and M. Walker, "Modelling protagonist goals and desires in first-person narrative," in *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2017)*, 2017.

[136] P. Rajendran, D. Bollegala, and S. Parsons, "Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews," in *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, 2016.

[137] D. Rao and D. Ravichandran, "Semi-supervised polarity lexicon induction," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, 2009.

[138] H. Rashkin, M. Sap, E. Allaway, N. A. Smith, and Y. Choi, "Event2Mind: Commonsense inference on events, intents, and reactions," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018.

[139] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019): Tutorials*, 2019.

[140] R. Rudinger, P. Rastogi, F. Ferraro, and B. Van Durme, "Script induction as language modeling," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, 2015.

[141] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, C. F. Baker, and J. Scheffczyk, *FrameNet II: Extended theory and practice*, 2016.

[142] E. Şahin, M. Cakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, pp. 447–472, 2007.

[143] I. Saito, K. Nishida, H. Asano, and J. Tomita, "Commonsense knowledge base completion and generation," in *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, 2018.

[144] A. Sancheti and R. Rudinger, "What do large language models learn about scripts?" in *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, 2022.

[145] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja *et al.*, "Multitask prompted training enables zero-shot

task generalization," *arXiv preprint arXiv:2110.08207*, 2021.

[146] M. Sap, R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi, "Atomic: An atlas of machine commonsense for if-then reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2019)*, 2019.

[147] R. C. Schank, "The primitive ACTs of conceptual dependency," in *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, 1975.

[148] ——, "Conceptual dependency: A theory of natural language understanding," *Cognitive Psychology*, vol. 3, no. 4, pp. 552–631, 1972.

[149] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1977.

[150] R. Schwartz, M. Sap, I. Konstas, L. Zilles, Y. Choi, and N. A. Smith, "The effect of different writing tasks on linguistic style: A case study of the roc story cloze task," *arXiv preprint arXiv:1702.01841*, 2017.

[151] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 2016.

[152] L. Shi and R. Mihalcea, "Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2005.

[153] E. Shutova, "Models of metaphor in NLP," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, 2010.

[154] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[155] K. K. Singh, S. Divvala, A. Farhadi, and Y. J. Lee, "Dock: Detecting objects by transferring common-sense knowledge," in *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, 2018.

[156] P. Singh, "The public acquisition of commonsense knowledge," in *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 2002.

[157] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. Li Zhu, "Open mind common sense: Knowledge acquisition from the general public," in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE: Confederated International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings*. Springer, 2002, pp. 1223–1237.

[158] A. Singhal *et al.*, "Introducing the knowledge graph: Things, not strings," *Official Google Blog*, vol. 5, no. 16, p. 3, 2012.

[159] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2017)*, 2017.

[160] S. Storks, Q. Gao, and J. Y. Chai, "Recent advances in natural language inference: A survey of benchmarks, resources, and approaches," *arXiv preprint arXiv:1904.01172*, 2019.

[161] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[162] M. Suhail and L. Sigal, "Mixture-kernel graph attention network for situation recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, 2019.

[163] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "Ernie: Enhanced representation through knowledge integration," *arXiv preprint arXiv:1904.09223*, 2019.

[164] S. Swayamdipta, S. Thomson, C. Dyer, and N. A. Smith, "Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold," *arXiv preprint arXiv:1706.09528*, 2017.

[165] S. Swayamdipta, S. Thomson, K. Lee, L. Zettlemoyer, C. Dyer, and N. A. Smith, "Syntactic scaffolds for semantic structures," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, 2018.

[166] H. Takamura and J. Tsujii, "Estimating numerical attributes by bringing together fragmentary clues," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2015)*, 2015.

[167] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," *arXiv preprint arXiv:1811.00937*, 2018.

[168] A. Talmor, O. Tafjord, P. Clark, Y. Goldberg, and J. Berant, "Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge," in *Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[169] P. P. Talukdar and F. Pereira, "Experiments in graph-based semi-supervised learning methods for class-instance acquisition," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, 2010.

[170] J. Tiedemann, "The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT," in *Proceedings of the Fifth Conference on Machine Translation*, 2020.

[171] T. H. Trinh and Q. V. Le, "Do language models have common sense?" 2019.

[172] Y. Tu and D. Roth, "Learning English light verb constructions: Contextual or statistical," in *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, 2011.

[173] J. Utt, A. Lenci, S. Padó, and A. Zarcone, "The curious case of metonymic verbs: A distributional characterization," in *Proceedings of the IWCS 2013 Workshop Towards a Formal Distributional Semantics*, 2013.

[174] E. M. Voorhees *et al.*, "The trec-8 question answering track report." in *Trec*, vol. 99, 1999, pp. 77–82.

[175] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[176] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," in *International Conference on Machine Learning (ICML 2022)*, 2022.

[177] S. Wang, G. Durrett, and K. Erk, "Modeling semantic plausibility by injecting world knowledge," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, 2018.

[178] N. Weir, A. Poliak, and B. Van Durme, "Probing neural language models for human tacit assumptions," *arXiv preprint arXiv:2004.04877*, 2020.

[179] D. S. Weld and J. De Kleer, *Readings in qualitative reasoning about physical systems*. Morgan Kaufmann, 2013.

[180] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, 2011.

[181] R. Wilensky, "Understanding goal-based stories," Ph.D. dissertation, Yale University, 1978.

[182] Y. Wilks, "A preferential, pattern-seeking, semantics for natural language inference," *Artificial Intelligence*, vol. 6, no. 1, pp. 53–74, 1975.

[183] W. A. Woods, "What's in a link: Foundations for semantic networks," in *Representation and Understanding*. Elsevier, 1975, pp. 35–82.

[184] Q. Wu, P. Wang, C. Shen, A. Dick, and A. Van Den Hengel, "Ask me anything: Free-form visual question answering based on knowledge from external sources," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016.

[185] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[186] Y. Xing, Z. Shi, Z. Meng, G. Lakemeyer, Y. Ma, and R. Wattenhofer, "KM-BART: Knowledge enhanced multimodal BART for visual commonsense generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL 2021)*, 2021.

[187] F. F. Xu, B. Y. Lin, and K. Zhu, "Automatic extraction of commonsense LocatedNear knowledge," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018.

[188] B. Yang and T. Mitchell, "A joint sequential and relational model for frame-semantic parsing," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 1247–1256.

[189] Y. Yang, L. Birnbaum, J.-P. Wang, and D. Downey, "Extracting commonsense properties from embeddings with limited human guidance," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018.

[190] M. Yatskar, V. Ordonez, and A. Farhadi, "Stating the obvious: Extracting visual common sense knowledge," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, 2016.

[191] M. Yatskar, L. Zettlemoyer, and A. Farhadi, "Situation recognition: Visual semantic role labeling for image understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016.

[192] Z. X. Yong and T. T. Torrent, "Semi-supervised deep embedded clustering with anomaly detection for semantic frame induction," in *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, 2020.

[193] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[194] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[195] H. Zhang, H. Ding, and Y. Song, "Sp-10k: A large-scale evaluation set for selectional preference acquisition," *arXiv preprint arXiv:1906.02123*, 2019.

[196] S. Zhang, X. Liu, J. Liu, J. Gao, K. Duh, and B. Van Durme, "Record: Bridging the gap between human and machine commonsense reading comprehension," *arXiv preprint arXiv:1810.12885*, 2018.

[197] L. Zhongyang, D. Xiao, and L. Ting, "Constructing narrative event evolutionary graph for script event prediction," in *In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 2018.

[198] X. Zhu and Z. Ghahramani, “Learning from labeled and unlabeled data with label propagation,” Carnegie Mellon University, Tech. Rep., 2002.

[199] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, 2003.