Use Defines Possibilities: Reasoning about Object Function to Interpret and Execute Robot Instructions

Mollie Shichman[1], **Claire Bonial**[2], **Austin Blodgett**[2], **Taylor Hudson**[3], **Francis Ferraro**[4], **Rachel Rudinger**[1]
[1] University of Maryland, College Park, [2] Army Research Lab
[3] Oak Ridge Associated Universities, [4] University of Maryland, Baltimore County
mshich@umd.edu, claire.n.bonial.civ@army.mil,
ferraro@umbc.edu, rudinger@umd.edu

Abstract

Language models have shown great promise in common-sense related tasks. However, it remains unseen how they would perform in the context of physically situated human-robot interactions, particularly in disaster-relief scenarios. In this paper, we develop a language model evaluation dataset with more than 800 cloze sentences, written to probe for the function of over 200 objects. The sentences are divided into two tasks: an "easy" task where the language model has to choose between vocabulary with different functions (Task 1), and a "challenge" where it has to choose between vocabulary with the same function, yet only one vocabulary item is appropriate given real world constraints on functionality (Task 2). DistilBERT performs with about 80% accuracy for both tasks. To investigate how annotator variability affected those results, we developed a follow-on experiment where we compared our original results with wrong answers chosen based on embedding vector distances. Those results showed increased precision across documents but a 15% decrease in accuracy. We conclude that language models do have a strong knowledge basis for object reasoning, but will require creative fine-tuning strategies in order to be successfully deployed.

1 Introduction

When it comes to using robots in disaster-relief scenarios such as search-and-rescue, it is essential that a robot can interpret and execute an instruction based on its current understanding of the objects detected in its environment. For example, in order to *Enter the building,* the robot should know to search for entrance points, such as doors and windows. Similarly, to *Scan the second floor,* the robot must be able to find appropriate ways to get to the second floor, such as stairs. Finally, to *Use the outlet to check for power,* the robot must know how outlets are used. Essentially, the robots need to know an object's function(s) in order to complete envisioned interactions.

Envisioned interactions are a multi-modal approach to responding to natural language instructions. For this paper, we assume that various sensors and computational systems, such as LIDAR, motion, or camera sensors, have taken care of identifying the objects in a scene. This information is passed to a language based world model, which deduces which, if any, of the objects perceived are relevant to the instruction based on the objects capabilities. This information would then be passed on to a lower-level policy-planning tool. An envisioned interaction that this research supports is depicted in Figure 1.

To understand the possibilities for executing a natural language instruction within the current environment, the robot requires *apriori*, commonsense knowledge of the objects in the environment. In particular, knowledge of object function is critical for interpreting natural language instructions in physically situated disaster-relief tasks. Given that such tasks are dynamic and dangerous, a robot should be able to accept unconstrained natural language (as opposed to placing a cognitive burden on the rescue worker to use a robot's controlled language). We hypothesize that a large language model (LM) would be uniquely equipped to handle this challenging task of supporting commonsense reasoning about an object's function for situated natural language understanding (NLU), due to the LM's latent world knowledge (Petroni et al., 2019).

The contributions of this paper include:

1. The development of a dataset of objects, found to be relevant to disaster-relief scenarios, with their functions established in terms of PropBank rolesets (Section 2);

2. The creation of an LM evaluation set of sentences that probe the model for its knowledge

Proceedings of the 15th International Conference on Computational Semantics pages 284–292
June 21–23, 2023.

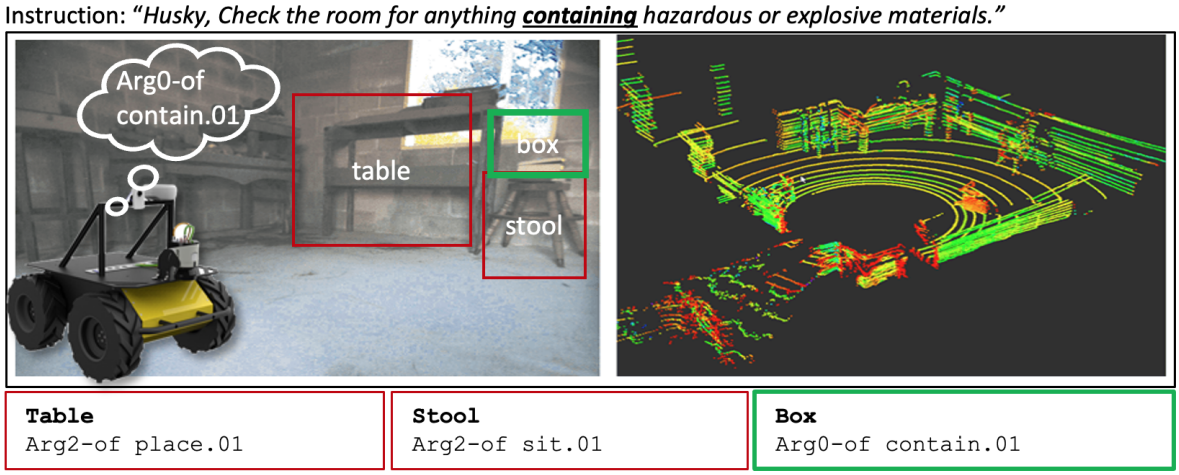

Figure 1: Envisioned interaction in which understanding and executing the instruction are supported by reasoning about objects in the environment detected via visual sensors (left) and LIDAR sensors (right). Given an instruction to look for a container of materials, the functions of detected objects with labels "table," "box," and "stool," can be compared against the containment function, represented by the PropBank role and roleset, "Arg0-of contain.01." Here, only "box" has the appropriate function, prompting further exploration of contents of the box.

of those object functions in both an "easy" task (Task 1) and a "challenge" (Task 2) (Section 3.1), and the augmentation of Task 1 for a follow-on evaluation (Section 3.2);

3. DistilBERT (Section 4) evaluation results (Section 5) with suggestions for future improvements informed by related work (Sections 6, 7).

We will make our object function dataset and cloze-sentence evaluation set available upon request.

2 Object Function Background and Dataset

PropBank (Palmer et al., 2005) is a semantic role labeling framework that provides a lexicon of event "rolesets," where each corresponds to a particular sense of a verb, eventive noun, or relational adjective. Each sense is described in terms of its set of participant roles, captured as argument numbers "ARG" 0-5, or as "ARG-M" modifier or adjunct arguments. In addition to the lexicon, PropBank provides a large corpus of annotated data where each relation is marked up with its sense in the lexicon and the arguments are marked for their semantic role with respect to that sense roleset. This lexicon is also used in the annotated corpus of Abstract Meaning Representation (AMR) (Banarescu et al., 2013). The standard roles are ARG0, which corresponds to Dowty's prototypical Agent, and ARG1, which corresponds to the prototypical Patient (Dowty, 1991). The corresponding semantic roles of the other, higher-numbered ARGs 2-5 are verb specific. ARG-Ms, which can theoretically modify or accompany any verb, include roles such as INSTRUMENT and PATH.

By leveraging the PropBank lexicon and corpus to establish that ladders and stairs fulfill the same role semantically (as the ARG1 for *climb.01*), we are able to derive a set of objects that have the same functionality (ways to climb between floors of a building).[1]Essentially, using Propbank is a pre-existing method of establishing commonalities between objects' functions. For example, Propbank allows us to group barrels, boxes, crates, and cabinets together because they all are ARG0 of the Propbank sense *contain.01*

While alternative resources that encode object functionality do exist, such as the Suggested Upper Merged Ontology (SUMO) (Niles and Pease, 2003), which includes axioms and object definitions indicating function, we found that PropBank provided a data-driven approach for us to develop a ground truth of each object's functionality as well as an elegant way of encoding and representing that function, for example as ARG1 of climb-01. This semantic representation of function thus fits with broader NLU that leverages the PropBank and Abstract Meaning Representation (AMR) (Banarescu et al., 2013) for a distillation of unconstrained natural language instructions into action primitives and their parameters, executable by a

[1]See climb.01 roleset: https://propbank.github.io/v3.4.0/frames/climb.html#climb.01

robot. Object function is therefore encoded in the same way as the natural language instructions that might reference the object or desired functionality. For example in Figure 1, the instruction would be parsed into AMR, abstracting the target object which would be a thing that is an ARG0-of contain.01. Then, the objects currently detected, localized, and labeled in the environment using the robot's sensors would be evaluated for which object had the matching function of ARG0-of contain.01. We created a vocabulary of about 280 objects mentioned in a human-robot dialogue corpus (Marge et al., 2017). These dialogues were previously collected via wizard-of-oz experimental interactions between people and remotely located robots in a search and exploration task, which is similar to our target domain of robotic exploration for disaster relief. Informed by existing PropBank and AMR corpora, an annotator then decided the best role for each vocabulary item given an appropriate PropBank sense. For instance, the word *crate* was assigned the PropBank ARG0 role of the *contain.01* sense—indicating it is the container holding some kind of contents. After one annotator made initial judgements, two other annotators familiar with PropBank and AMR reviewed the annotation to validate or offer alternative labels for vocabulary whose PropBank annotations were more difficult to surmise.

3 LM Evaluation Dataset

With the objects labelled with the appropriate Propbank sense-role pairing to signify their functionalities, we needed to develop a method of zero-shot testing a language model. For this methodology, it was important that we develop an understanding both of the language model's capabilities and how a small group of expert human annotators could be skewing the results beyond their particular writing styles. This led to two rounds of evaluation data generation: one with a manually developed answer set, and one with an answer set based on distances within LM vector space.

3.1 Manually Developed Sentence and Answer Set

We wanted to analyse both the LM's ability to differentiate between objects with different functions (Task 1) and between objects with the same function (Task 2), so we designed two different tasks. For both, we generated cloze sentences that express the need for a particular functionality or affordance, where the correct answer is one of our object vocabulary items that offers that functionality (according to the function annotations described in Section 2). The LM's task was to pick the correct word from a short list of possible answers. Providing a short list of answers was both inspired by the Winograd Schema Challenge (Levesque et al., 2012) and because a robot would be faced with a set of recognized and labeled objects in its environment to choose from in a given disaster-relief scenario.

For Task 1, annotators wrote sentences such that all words with the same function can reasonably fill in the blank. For instance, in the sentence *Go check if there's anything suspicious inside that BLANK*, the blank can be filled by any word denoting an object whose function is a container, be it a barrel or a cabinet. In Task 1, two wrong options were also presented; these did not share the function of the right answer and were arbitrarily chosen by the annotator from the rest of the vocabulary list. One sentence was written for each function. If more than one vocabulary term had the same function, the same sentence would be used multiple times, but the correct answer would be changed so that each word with the same function was represented in the evaluation set. This was done to see if a LM was consistent in correctly choosing objects with the same function. The sentences were written fairly explicitly so that only the word's intended function could be reasonably inferred by a human reader, as we had words that could serve multiple functions, like *stairs*, which could fall under *ascend.01* PATH or *descend.01* PATH.[2] A sample of Task 1 sentences from one author/annotator is given below. The LM must choose which of the answer choices is the most likely filler of the masked position.[3]

(1) I need to see from higher up, so I'm going up the [MASK].

Choices ladder, cushion, tomato

Correct ladder

(2) I need to see from higher up, so I'm going up

[2]While a qualitative analysis of the results did not show any evidence of polysemous words causing errors, we are uncertain how many vocabulary items are polysemous and what that effect may have on our results.

[3]From an implementation perspective, we used the following format: SENTENCE with [MASK] ||| ANSWER CHOICES ||| CORRECT ANSWER.

the [MASK].

Choices stairway, cushion, tomato
Correct stairway

(3) The [MASK] will keep the horse from running out of the pen.

Choices mop, barrier, bucket
Correct barrier

(4) The roof collapsed when the flimsy [MASK] failed to support its weight.

Choices curtain, lamp, column
Correct column

Note that the answer vocabulary is based upon objects mentioned in the human-robot collaborative exploration corpus, and therefore relevant to robotic exploration tasks, even if the sentences are not instructions per se. By not limiting the annotators to writing instructions only, we allowed for more use-cases given the object's function. For example, here are three sentences given the function of *contain.01 ARG0*.

1. I was getting ready to move, so I put all of my belongings into a [MASK].
2. Go check if there's anything suspicious inside that [MASK].
3. I need to hold my collection of cups for safekeeping, so I'm going to use a [MASK].

Each sentence works for any objects that can contain, but they each highlight a unique aspect of containing that would be important for a robot to recognize.

For Task 2, we narrowed our focus in order to study how LMs can leverage commonsense knowledge to differentiate between items with the same function. For our initial evaluation, we chose two functions from our dataset that contained the most unique objects within them: facilitating transport (objects listed with this function include *car, boat, bike*) and containment (objects listed include *jug, luggage, cup*). Within each function, we wrote sentences that would be true for one object with the same functionality but not another. As an example, the LM could choose between *ladder* and *stairs* to fill in the blank for *I need to get to the second floor, so I'm going to move the BLANK to that window*. Both serve the function of climbing, but they are not interchangeable because ladders are portable and stairs are not. We generated all possible pairings of objects within our chosen functions and randomly selected the pairings for sentence generation. More details about the sentence data can be found in Table 1 and a sample from one annotator for the transportation function is given here:

(4) I'm trying to get my legs in shape, so I take my [MASK] to school each day.

Choices bicycle, boat
Correct bicycle

(5) My husband's going green so he takes his [MASK] everywhere he needs to go.

Choices bicycle, car
Correct bicylce

(6) Today you really need air conditioning, so you decide to take the [MASK] to get to the office.

Choices bicycle, car
Correct car

(7) She couldn't afford any gas, so she had to ride her [MASK] to the next village over.

Choices bicycle, motorcycle
Correct bicycle

Note that the real-world knowledge required to determine the correct answer for Task 2 we hypothesized to be fairly nuanced—a connection between biking and *getting legs in shape*, or *going green*, or *NOT being able to afford gas*, for example.

3.2 Answer Sets from Embedded Vectors

After an initial analysis of the results of Task 1, we noted that the performance across "documents," where each document is the set of evaluation sentences written by a single annotator, varied substantially (as we will describe in greater detail in Section 5). This prompted us to consider where this variation was coming from. Each document was intended to evaluate the LM's knowledge of the functionality of the same set of objects, so this was variance outside of what could be concluded to be related to commonsense knowledge of object functionality. We only had three annotators, which has been shown to introduce bias (Geva et al., 2019). As each annotator both authored sentences and selected the sentence's wrong answers, we hypothesized that both factors likely add bias to our

results. As a first step to reduce inadvertent variance stemming from wrong answer choices, we elected to experiment with different methods of choosing wrong answers for Task 1 to see how the wrong answers affected the results of our experiments. Specifically, we decided to compare the results with the manually chosen wrong answers for Task 1 with a more technical procedure in which we selected the wrong answers based on the cosine distance between vectors taken from the LM's result of encoding each individual vocabulary term. While the embedded vectors of individual words differ from the embedded vector the word takes within an encoded BERT sentence, we decided this was a reasonable approximation that also took into account the limited onboard computing power a robot would have for our task.

Go check if there's anything suspicious inside that ____.
Task 1: ***A) barrel*** ✔ B) staircase C) road
Task 1e: A) barrel ***B) pipe*** ✗ C) bucket
There's a fire! Does anyone know how to put it out with a ____?
Task 1: ***A) hydrant*** ✔ B) boat C) counter
Task 1e: A) hydrant B) separator ***C) gas pump*** ✗

Figure 2: Two examples of the effect different answer choices for task 1 vs task 1e. The answers chosen for closeness by vector distance are often have similar functions (carry vs. contain) or potentially related within a conceptual domain (gas pump vs. hydrant), making Task 1e more challenging.

We ran several experiments at closer and further cosine distances to test the hypothesis that the LM would choose more wrong answers if they had a closer cosine distance to the correct answer. We named this "Task 1e," or Task 1-encoded vectors. For each right answer, we compiled a subset of valid distractors from our original vocabulary list, then chose the wrong answers by their ranking in our query. Examples comparing sentences and answers for Task 1 and 1e are shown in Figure 2. This approach does not account for any changes in density within the vector space. However, for all experiments the standard deviation of distances remained fairly uniform. This led us to believe that the ranked distances were all similar enough that the comparison between functions is still fair.

4 Experimental Setup

We used Huggingface's pipeline class with the fill-mask task and the DistilBERT uncased model. We chose DistilBERT because it is lightweight while having very similar accuracy to the full base BERT model (Sanh et al., 2019). This allows the model, theoretically, to be loaded directly onto the robot platform, keeping its space to a minimum without sacrificing too much accuracy. To calculate the vector embedding's cosine distances, we followed in BERT-as-a-service's footsteps: we took the second to last layer of DistilBERT to represent each vocabulary term (McCormick and Ryan, 2019). We used Sci-kit Learn's implementation of a KD-Tree to store the resulting vectors (Pedregosa et al., 2011). All experiments were run with Pytorch and all scores were put into log space (Paszke et al., 2019).

	Task 1	**Task 2**
Sentences	608	236
Objects	183	21
Functions	65	2
LM Accuracy	**81.5%**	**79.7%**
Acc. Range	**22.8%**	**15.0%**

Table 1: Size and shape of the data, as well as DistilBERT's average accuracy for Task 1 and Task 2 and the range in its accuracy across documents.

Multi-token Vocabulary Terms One challenge we faced was how to fairly compare the scores of single-token vocabulary terms as opposed to multi-token vocabulary terms, since the WordPiece tokenizer used by DistilBERT can potentially break words into subwords. To solve this problem, we adapt the sentence level scoring scheme of pseudo-log likelihood from Salazar et al. (2020) when vocabulary items have multiple tokens. Specifically, for tokens $t_1 \dots t_n$ that make up word W with T_j tokens before the mask and T_k tokens afterwards, where j and k are both natural numbers, we calculate the probability as shown:

$$log(p(t_1|T_j, T_k)) + log(p(t_2|T_j, t_1, T_k)) \dots + log(p(t_n|T_j, t_1, t_2, \dots t_{n-1}, T_k))$$

We found that normalizing the scores by the number of tokens improved accuracy results. We hypothesize that this normalization reduced the LM's bias towards single token answers, but more experimentation is required to fully understand the effects of normalizing scores by token length.

5 Results and Discussion

5.1 Task 1 and Task 2

The accuracies for Task 1 and Task 2 were nearly identical, as shown in Table 1. This was somewhat

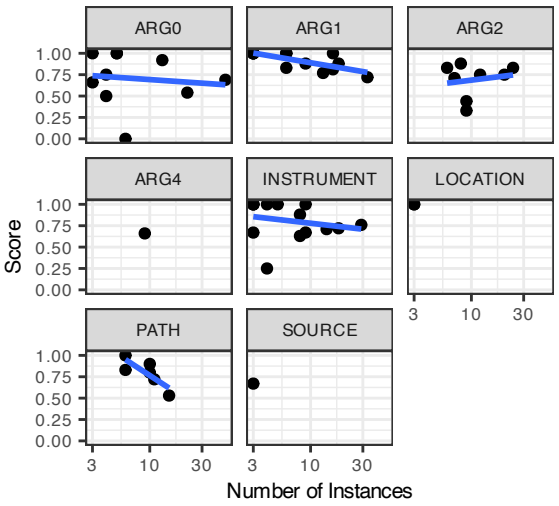

Figure 3: A breakdown of accuracy across PropBank roles for Task 1 by number of instances. Notably, accuracy decreases as the number of samples increases regardless of the role the vocabulary term plays in the sentence.

surprising, as we thought that DistilBERT would be more accurate when differentiating between words with different functions than within the same function, where we hypothesized more nuanced commonsense knowledge was required to recognize the correct answer. This could be from word co-occurrence probabilities. DistilBERT knows that *legs* is more likely to co-occur with *bicycle* than *boat*, so it doesn't necessarily need to do any reasoning. It's also possible that reporting bias played a role: annotators may have spent more time carefully differentiating between objects with similar functions than they do differentiating objects with significantly different functions because it is more self-explanatory to the reader what the latter differences are. Thus, the sentences for Task 2 may have inadvertently been more informative.

We also obtained DistilBERT's accuracy across each function. Some trends are immediately visible. First, regardless of the role the vocabulary term played in the sentence, the more sentences written for a specific function, the worse accuracy got. This can be seen in 3. Since so much expert knowledge was used when assigning the PropBank sense and role while deciding function, we do not believe this is because of labelling error. Rather, functions with fewer sentences tended to be more common, specific, and explicit than functions with many sentences. For instance, some functions that had only one or two sentences that scored well were

Ranked Distance	Accuracy	Accuracy Range
1st, 2nd	62.2%	15.4%
1st, 3rd	60.1%	15.2%
2nd, 3rd	66.5%	13.3%
2nd, 5th	67.2%	10.0%
6th, 11th	76.0%	9.1%
12th, 21st	77.1%	18.7%
26th, 36th	81.5%	7.3%

Table 2: Results for Task 1e. Ranked distances refer to the cosine distance from the wrong answers to the correct answer and are ranked by closeness to the correct answer, from 1st closest to 36th closest.

dig.01 ARG2 (which corresponded with shovel), rotate.01 ARG1 (which corresponded with wheel), and buttress.01 ARG0 (which corresponded with column). All of these items are strongly correlated with the functions. Larger categories that struggled more included contain.01 ARG0, whose vocabulary items ranged from cabinet to can, and occupy.01, whose terms ranged from car to barn. Since the annotators were writing sentences that worked with all vocabulary of the same function, the sentences with "larger" functions had to be more general and likely had fewer semantic clues for DistilBERT to utilize. This suggests that LMs have room to improve on more general cases for objects for our use case, including handling a wider variation in object function use.

Even though the results for Task 1 were strong, within the task there was a wide range in accuracy over each document, with 2 documents in the same task differing in accuracy by as much as 22%. We attributed this wide range to annotator bias (as mentioned in Section 3.2). While annotator bias is a given in a dataset with few sentence creators, we wanted to minimize as much bias as possible to ensure the LM was a sufficient basis for our ultimate use case of collaborative, disaster-relief communication. One clear place to eliminate bias was in the selection of wrong answers, motivating the development of Task 1e.

5.2 Task 1e, Embedding Distances

For Task 1e, we achieved our initial goal of reducing the range in accuracy over all documents for all experiments, as shown in Table 2. This demonstrates that the wrong answers chosen by sentence authors did have an impact on accuracy, as we had hypothesized. The overall accuracy ranges also show that the impact of manually selected wrong answers is overall positive. In other words, the

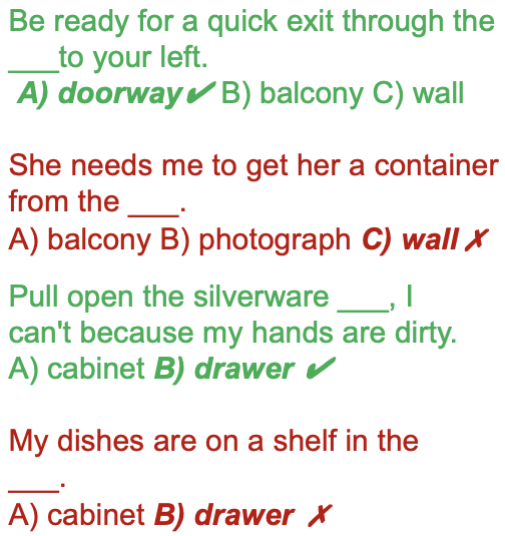

Figure 4: Example sentences that DistilBERT correctly (shown in green with check marks) and incorrectly (shown with a red *X*) answered from Tasks 1 & 2.

manually selected wrong answers in Task 1 were generally easier for the LM to eliminate than the wrong answers selected for all but the most distant wrong answer choices in Task 1e. The accuracy range also decreases as vocabulary terms get further away from the correct answer in vector space, demonstrating that the sentence alone does not give DistilBERT enough information to differentiate between the answers, and that it needs the answers choices to provide extra information for it to make a correct decision. We also examined the scores for each function as we did with Task 1, and we found that scores decreased rather evenly across the board, regardless of how many sentences were testing the function.

As we had hypothesized, the overall scores and the scores by function generally improved linearly as the wrong answers moved further away from the correct answer. However, when looking at individual documents and functions with wrong answers close to the correct answer, that linearity breaks down, and performance seems very dependent on the language choices of individual annotators. When examining the data qualitatively, it's often not clear from a linguistic perspective why DistilBERT assigned the probability it did. For instance, DistilBERT thought it was more likely that one would use a motorcycle to *catch their balance* than a rail, or even a television. It's also not immediately clear how the annotators writing styles are "easier" or "harder" for DistilBERT to work with. Other unclear examples can be seen in Figure 4 for both Task 1 and 2. We suspect that larger language models which utilyze larger vocabularies than DistilBERT would be more linguistically informed due to the increased data and training time, but we leave that to future work.

While the scores decreased significantly when going from annotator-selected wrong answers to ranked distance wrong answers, DistilBERT still scores far better than random and shows it does have a strong amount of knowledge on object functions. Overall, our expectations for DistilBERT's zero-shot knowledge were exceeded in both tasks. Nonetheless, given the high stakes of our application domain, we plan paths for improvements in future work (Section 7).

6 Related Work

We were inspired in our own research by Chen et al. (2022), who also test an LM's zero-shot knowledge with respect to physically situated settings. The authors' goal is to use LMs to help robots determine the type of room it is in for a given 3D scene. To test if LMs could be effective at this task, they automatically generate sentences from the template "The *r* often contains *o*", where *r* is a type of room and *o* is an object often found in that room. The authors ran their sentences through the masked LM BERT with the room masked to see how well BERT could predict the room based on the objects. The authors found that rooms with very specific items (bathrooms, bedrooms, kitchens) were easier to identify than rooms which had furniture that can be in many rooms (dining rooms, living rooms). This showed us the effects reporting bias can have on physical commonsense LMs and prompted us to research this for our own use case.

The ultimate goal of our research is to use LMs for robot policy planning with a strong understanding of the LM's decision-making process and embeddings, since high stakes situations demand accountability. Dipta et al. (2022) approach this task by creating linguistically informed embeddings within a custom encoder-compressor-decoder network. The network was trained to recognize the hierarchical nature of events by using frames from FrameNet (Baker et al., 1998) only partially describing said event. By injecting linguistically informed knowledge, while not requiring specific vocabulary to indicate that an event is occurring, Dipta et al. (2022) had strong performance with a reasonable explanation of what each part of the

neural network is doing.

In terms of planning with LMs, there are multiple interesting approaches. Driess et al. (2023) trained an LM, called PaLM-E to also accept image and continuous sensor data, as well as text. By encoding the non-text data into vectors that are the same size as a text vector, the model can complete a variety of tasks straight out of the box while also allowing for downstream fine-tuning. Notably, it can output plain text that can be interpreted as a robotic policy, though PaLM-E has to interpret on its own what a particular robot's capabilities are. More testing needs to be done to see if the robot can behave consistently, and the authors caution that it is not meant for long-term tasks. Another model made by Song et al. (2022) utilizes an upper level LM, in their case GPT, with some few-shot training for high-level policy planning. They separately designed a lower level model that handles the execution of movement and other low-level tasks. Importantly, if the lower level model can't execute a task, it can query the higher level model with the information it perceived about the environment for an updated policy. This enables it to handle long term, complex tasks. However, both of these models lack the explainable nature of Dipta et al. (2022) with its basis in linguistic theory.

7 Future Work and Conclusion

Given the overall success of these experiments, we have several avenues of future work. First, we want to test how different LMs perform on our dataset. While DistilBERT satisfied our theoretical computational constraints, there's a strong chance that newer and larger masked LMs will perform even better on our dataset. Testing on other LMs will also further solidify our dataset as a useful analysis tool for object-related common sense. We also want to do a more in-depth statistical analysis of how DistilBERT performed by function, perhaps grouping functions to get coarser granularity to understand which functions need the most fine-tuning for a LM to succeed.

With the recent advent of multi-modal LMs like PaLM-E and GPT-4 (OpenAI, 2023), our research interests are quickly shifting towards utilizing these models for grounded common-sense understanding. It is possible these may be more aware of physical limitations due to images (and in PaLM-E's case, robotic policy) in the training data. While these models do have some ability to explain their decision making process, there is much to discover in terms of the models' full capabilities. We are also interested in examining few-shot fine-tuning with syntactic and semantic information to improve both common-sense performance and the model's ability to explain itself. Our hope is that combining new multi-modal models with linguistic insight will make a more trust-worthy model that can be successfully deployed in disaster-relief missions.

We set out to discover if LMs can provide the type of *apriori*, commonsense knowledge of the functions of various objects, especially those deemed important to robot-based, disaster relief missions. This is important because this technology could lead to replacing humans with robots in dangerous scenarios that have little room for error. We systematically identified the function each object plays in our domain, then created two tasks to test the granularity of a LM's ability to differentiate between these functions. DistilBERT performed quite strongly on our tasks, validating our proof of concept. Even when removing the bias of human-generated wrong answers, we still obtained strong results indicating that DistilBERT has significant knowledge about our domain. We are finding new avenues to expand our research into using more advanced LMs in tandem with resources encoding linguistic knowledge to improve collaborative, physically situated human-robot dialogue.

Acknowledgments

We would like to thank the anonymous reviewers for their comments, questions, and suggestions. This material is based in part upon work supported by the National Science Foundation under Grant No. IIS-2024878. This material is also based on research that is in part supported by the Army Research Laboratory, Grant No. W911NF2120076, and by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Air Force Research Laboratory (AFRL), DARPA, or the U.S. Government.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

William Chen, Siyi Hu, Rajat Talak, and Luca Carlone. 2022. Extracting zero-shot common sense from large language models for robot 3d scene understanding.

Shubhashis Roy Dipta, Mehdi Rezaee, and Francis Ferraro. 2022. Semantically-informed hierarchical event modeling.

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A William Evans, Susan G Hill, and Clare Voss. 2017. Applying the wizard-of-oz technique to multimodal human-robot dialogue. *arXiv preprint arXiv:1703.03714*.

Chris McCormick and Nick Ryan. 2019. Bert word embeddings tutorial.

Ian Niles and Adam Pease. 2003. Linking lixicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In *Ike*, pages 412–416.

OpenAI. 2023. Gpt-4 technical report.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. 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)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2022. Llm-planner: Few-shot grounded planning for embodied agents with large language models.