# What Can a Generative Language Model Answer About a Passage?

**Douglas Summers-Stay** and **Claire Bonial** and **Clare Voss**
U.S. Army Research Laboratory
douglas.a.summers-stay.civ@army.mil

## Abstract

Generative language models trained on large, diverse corpora can answer questions about a passage by generating the most likely continuation of the passage in which the answer to a given question, as appended to the passage, is the most likely continuation of that passage. However, accuracy rates vary depending on the type of question asked. In this paper, we keep the passage fixed, and test with a wide variety of question types, exploring the strengths and weaknesses of the GPT-3 language model. We provide the passage and test questions as a challenge set for other language models.

## 1 Introduction

Generative language models produce likely text based on a context of other text. This process has a surprising number of useful applications, one of which is answering questions about a text passage. By training on text that contains (among other data) passages followed by questions and answers about the passage, and creating a context in which the answer to a question is the most likely continuation of the passage, better text prediction tends to result in better question answering.

However, there are many types of questions that could be asked about a passage, from direct questions about facts, to explorations of its themes, to imagining what would happen if something about the story was completely different. Depending on the type of question, the accuracy of answers can vary greatly. While some tests of the question-answering ability of language models have been run, it is difficult to separate out whether a question is answered incorrectly because of changes to the question, or changes to the passage, or both. Therefore, in this paper, we selected a single passage and asked a wide variety of questions about the passage. The response of a particular model to any one of these types of questions could be explored much more extensively on a larger dataset with more passages. Here, we are simply making a first survey of the possibilities.[1]

| **New York Times, 29 September 1973** |
|---|
| A 61-year old furniture salesman was pushed down the shaft of a freight elevator yesterday in his downtown Brooklyn store by two robbers while a third attempted to crush him with the elevator car because they were dissatisfied with the $1,200 they had forced him to give them.<br>The buffer springs at the bottom of the shaft prevented the car from crushing the salesman, John J. Hug, after he was pushed from the first floor to the basement. The car stopped about 12 inches above him as he flattened himself at the bottom of the pit.<br>Mr. Hug was pinned in the shaft for about half an hour until his cries attracted the attention of a porter. The store at 340 Livingston Street is part of the Seaman's Quality Furniture chain.<br>Mr. Hug was removed by members of the Police Emergency Squad and taken to Long Island College Hospital. He was badly shaken, but after being treated for scrapes of his left arm and for a spinal injury was released and went home. He lives at 62-01 69th Lane, Maspeth, Queens.<br>He has worked for seven years at the store, on the corner of Nevins Street, and this was the fourth time he had been held up in the store. The last time was about one year ago, when his right arm was slashed by a knife-wielding robber. |

Table 1: Full text of story in McCarthy (1990).

## 2 The Challenge Set Passage

In 1976, John McCarthy published the informal memo "An Example for Natural Language Understanding and the AI Problems It Raises" (McCarthy, 1990). The example was a short few paragraphs from a news article about a robbery, and a series of questions about the article asked and answered in natural language. The full text of the passage can be found in Table 1.

[1] To assess for statistical significance beyond this manual survey, the next phase of research will entail automating the generation of questions from type seed sets and the evaluation of question/answer pairs.

*Proceedings of the 3rd Workshop on Machine Reading for Question Answering* , pages 73–81
Nov 10th, 2021 ©2021 Association for Computational Linguistics

The term "natural language understanding" only seems to have been introduced to the field of AI in about 1973 or 1974 (Woods, 1973). With this 1976 memo, McCarthy was helping lay the foundations of what the field would mean for AI researchers. He called this newspaper story "my candidate for a target for a natural language understander," pointing out that a fictional story would bring up even more difficult to answer questions about the motivations of the author.

McCarthy thought it would make the problem simpler to have both the questions and answers *not* be in natural language, but be queries formulated in an artificial query language free of ambiguities. Parsing the questions does not seem to require much beyond what would already be needed to parse the story, but certainly natural language generation was not part of the problem as he conceived it at all.

As an early benchmark, it avoids the kinds of bias that come from working with modern systems, tailoring the kinds of questions we ask to what the models we are working with can or can't do, depending on the researcher's motivations. It is small and simple enough to fit within the prompt of GPT, but rich enough that it allows tests for many different aspects of "understanding."

## 3 Background

We present views on what would be needed to address McCarthy's challenge, an overview of the GPT-3, and prompt-based learning.

### 3.1 Early Views of NLU Architecture

The paper from which the passage was taken contain's McCarthy's thoughts on how one might build a system to answer the questions he posed. It is instructive to compare the architecture of the GPT models tested in this paper with what McCarthy and, later, Erik Mueller believed would be required to solve such problems (1999).

McCarthy proposed that the Natural Language Understanding problem be solved by a system with the following components:

"1. A 'parser' that takes English into ANL [Artificial Natural Language].
2. An 'understander' that constructs the 'facts' from a text in the ANL.
3. Expression of the 'general information' about the world that could allow getting the answers to the questions by formal reasoning from the 'facts' and the 'general information.' The "general information" would also contain non-sentence data structures and procedures, but the sentences would tell what goals can be achieved by running the procedures. In this way, we would get the best of the sentential and procedural representations of knowledge.
4. A 'problem solver' that could answer the above questions on the basis of the 'facts.'"

Mueller (1999), responding to McCarthy's paper, proposed that a system for answering the questions in the passage would require the following steps:

"1. Feed input text to text agents for recognizing entities such as names and places.
2. Perform lexical analysis on the input.
3. Use part-of-speech taggers and word sense disambiguators to reduce possibilities.
4. Feed textual entities and lexical entries to a partial syntactic parser.
5. Feed syntactic parse fragments to a semantic parser.
6. Feed semantic parse fragments to a collection of understanding agents.
7. Build understanding agents for multiple realms including the physical realm, devices, human needs and goals, emotions, and mental states.
8. Design and implement mechanisms for understanding agents to negotiate a shared interpretation and renegotiate that interpretation as each input is received.
9. Build B-Brain mechanisms for controlling processing by understanding agents.
10. Adapt and extend existing parsers and lexicons.
11. Evolve existing commonsense databases by adding knowledge as needed.
12. Build links between existing resources, allowing multiple resources to be used."

The neural approaches in this paper are missing all of these components. However, there is some evidence that deep neural language models may rediscover a similar NLP pipeline (Tenney et al., 2019; Li et al., 2021).

The knowledge-base-centric approaches suggested by McCarthy and Mueller would guarantee that if the background knowledge, passage, and questions were encoded as logical statements correctly, the inference would be valid and the answers true. When using a language model to answer the questions, we have no such guarantees.

### 3.2 GPT-3

GPT stands for "Generative Pre-trained Transformer." It is a deep neural network with the transformer architecture, trained on a large general text corpus, that generates text as output, given a text prompt. Both the input and output are represented as "tokens": either common words or parts of words that can represent any unicode string. There are 50,000 possible tokens, and at each step, GPT generates a probability distribution over these tokens. How this probability distribution is sampled depends on various parameter settings, but for nearly all results in this paper, we simply select the most probable token.

The Davinci model of GPT-3 is the most powerful of this language model (LM) family, at 175 billion parameters for the Davinci version. GPT-3 has set records for accuracy on several question-answering and common-sense tasks (Brown et al., 2020), which made it a reasonable choice for the model best able to answer these questions correctly at present. In addition to being trained on terabytes of general text data from the web, the *instruct* variants, such as Davinci-instruct-beta (GDIB), were also specially trained on examples of question-answering, in a process known as "finetuning." This finetuning causes the model to have a bias towards expecting text to fall into a pattern of a short section of text, followed by questions and answers about that text.

We also did limited testing of the other GPT-3 models. These are referred to as Ada, Babbage, Curie. Although the sizes of these models have not been made public, indirect evidence (Gao, 2021) suggests that they are 350M, 1.3B, and 6.7B parameters respectively.

### 3.3 Prompt-based Learning

Stepping back for a moment, it is worth pointing out just how distinctive applying the LM-based approach to question-answering is in the context of machine learning methods and where prompt engineering fits in to this task. In traditional supervised learning, in order to train the parameters of the model, supervised data, i.e., pairs of inputs and outputs, are needed to characterize the task that the model will perform. The crucial issue for many tasks, however, is that the large amounts of supervised data needed for training these models may not be available. One way around this requirement when the datasets involve text is to leverage the text prediction capability of language models. While it is beyond the scope of this paper to describe the training of LMs specifically for prompt-based applications,[2] we note here the format of the prompts that we used in running the questions of the challenge dataset we have constructed:

"Read the passage below and answer the questions based on what you read."
passage
"Q. "
the question
"A."

The first element of the prompt was an instruction, altered systematically in our experimentation, as described in Section 4.1.1. The second element was the text passage verbatim (Table 1). The third element was the indicator of the upcoming question. The fourth element was the question verbatim. And the final element was the indicator of an answer to come.

## 4 Approach: Challenge Set Development

We construct our challenge dataset by beginning with the passage and questions from McCarthy, and building upon this with questions that further probe hypothesized strengths and weaknesses of GPT-3.

### 4.1 Questions from McCarthy and Answers from GPT-3 Davinci-instruct-beta

The questions were presented as a numbered list in McCarthy's paper. We presented the text of the article that McCarthy extracted, followed by each question individually (so previous questions and answers are not included in the prompt). The question was preceded by the letter "Q." to indicate that it is a question and followed by the letter "A." on a new line to indicate that an answer is expected. This is the most common use pattern for Davinci-instruct-beta.

We have set the temperature parameter to 0 except where otherwise noted, so that the most likely token is selected at each step. This also makes the results more reproducible. Temperature 0 has an increased tendency toward repetition in long (multiple sentence) answers and tends to produce somewhat blander output, but for this task it is probably the most appropriate.

The full set of questions and answers are in the Appendix B. McCarthy did not answer a few of his own questions, probably because the answer

[2] For details see Liu et al. (2021).

was obvious, and he followed up with a related question.

### 4.1.1 Summary Analysis on McCarthy Questions, Answers

Of McCarthy's 25 questions, GDIB clearly got four wrong, with perhaps another five arguable. We tested these same questions on all six GPT-3 models, with three variations on the prompt:

1. The passage followed by "Q." then the text of the question, and then "A." These results are listed in Appendix D.
2. Same as 1. above but preceded by the text "Read the passage below and answer the questions based on what you read."
3. Same as 1. above but preceded by the text "Read the passage below and answer the questions based on what you read. If there is not enough information to answer, say so."

| Model | #1 | #2 | #3 |
|---|---|---|---|
| Ada | 12 | 12 | 13 |
| Babbage | 15 | 18 | 17 |
| Curie | 17 | 17 | 18 |
| Davinci | 15 | 18 | 18 |
| Curie-instruct-beta | 18 | 17 | 18 |
| Davinci-instruct-beta | 21 | 20 | 19 |

Table 2: Number of correct answers out of 25 on each model and for each prompt variation.

See Table 2 for a summary. This is a small dataset and the grading is not exact, but in general we see that including instructions in the prompt helped the non-instruct models and had little effect on the instruct models (probably because it was only directing them to do what they had already been fine-tuned to do); and that the Babbage, Curie, and Davinci models did not differ much in their ability to correctly answer these questions, though Ada did worse and the instruct models did a little better.

GPT's successes and failures on these questions hinted at types of questions to try to further test the model's ability. We only tested these questions on GDIB, the best-performing model, without instructions in the prompt.

## 4.2 Added Challenge Set Questions

The questions McCarthy asked covered a broad range of topics, and probed several different abilities of the system. We created additional questions to more thoroughly cover different kinds of questions that systems might be capable of answering.

This approach provides us with a way to further organize the categories of challenge set questions and begin to spell out, for example, a few parallel hypotheses with respect to these questions and the abilities of GPT-3:

**Knowledge Sources** Questions pertaining to explicit information within a single sentence in the prompt are more likely to be answered correctly than to explicit information that is distributed across sentences in the passage in the prompt.

**Knowledge Types** Questions pertaining to a single common named entity are more likely to be answered correctly than those pertaining to sequencing of several events and specific entities involved in those events.

**Reasoning Requirements** Questions with logical negation are more likely to be answered incorrectly than their counterpart without negation.

Outside of these basic hypotheses, we view this research as exploratory to further probe strength and weakness areas of GPT-3 generally, with the three dimensions allowing us to pinpoint these areas more precisely.

### 4.2.1 Knowledge Source

Questions requiring only factual knowledge to answer were almost always answered correctly:

- Questions about the facts of the article that require little external knowledge (e.g., ***How old was Mr. Hug?***)
- Questions that require combining facts of the article with some other commonly known information (e.g., ***What country did the events take place in?***)

These questions would still be challenging for a system with a knowledge base (as opposed to a LM), because mapping the natural language questions to the logical representation and mapping the answers to natural language is not trivial. However, these steps being handled, the content of the questions would be a straightforward retrieval.

Questions requiring mainly linguistic knowledge were also usually answered correctly. These questions mainly involve interpreting an ambiguity in language correctly based on the surrounding context, such as defining a word with the right sense (e.g., ***In the phrase "he was badly shaken" what does the word "shaken" mean?*** *The word "shaken"* *means to be very frightened*).[3]

It has been argued (Saba, 2020) that certain linguistic phenomena should be especially difficult for natural language understanding due to the missing text phenomenon, where the speaker leaves out certain information from speech because the listener already knows the information, making it impossible to learn from text alone. However, our results indicate that for many instances of this phenomenon, this is simply not the case. Apparently, much of the the common-sense information is available to the model in some form that it is able to make use of. We tested the following types of missing-text inference:

- lexical ambiguity (***How was Mr. Hug treated at the hospital?*** *He was treated for scrapes on his left arm and for a spinal injury.*)
- hidden relations (***When it says, “the car stopped 12 inches above him”, what did the car stop doing?*** *The car stopped moving.*)
- quantifier scope (***Is every Seaman’s Quality Furniture store on Livingston Street?*** *No, there are many stores in the chain.*)
- prepositional ambiguity (***What does “for” mean in the sentence “he was treated for scrapes”?*** *“For" means “in order to treat."*)
- metonomy (***In the phrase “a third attempted to crush him,” what does the word “third” refer to?*** *The third robber.*)
- metaphor (***In the phrase “he was pinned in the shaft” what does the word “pinned” mean?*** *The word “pinned" means to be held in place.*)
- compound nominals (***What does a “freight elevator” have to do with freight?*** *A freight elevator is a large elevator that is used to transport goods.*)

While the model does seem to contain this information, depending on how the question is phrased it might be inaccessible to the model. For example, ***A “shoe store” sells shoes. A “book store” sells books. What does a “downtown Brooklyn store” sell?*** *** A “downtown Brooklyn store” sells “Brooklyn.”*[4] The model sometimes gets questions of this form right and sometimes wrong. This example demonstrates the effect that careful prompt engineering can have—the prompt examples seem to sway the system towards an analogous, but in this case incorrect, answer.

### 4.2.2 Knowledge Type

Beyond simple factual questions are questions that test comprehension or understanding. One definition of “understanding” is “to grasp explanatory and conceptual connections between various pieces of information involved in the subject matter in question” (Kvanvig, 2015). In this sense, GDIB is able to solve some “understanding” tasks.

**Theory of Mind** GDIB is able to reasonably answer questions about what characters were thinking and feeling (e.g., ***Why did Mr. Hug give the robbers $1200?*** *He was afraid they would kill him*). These are “theory of mind” questions, intended to probe the ability to ascribe thoughts, beliefs, and feelings from one’s own for a variety of reasons, including the other’s current knowledge. For a person to answer these questions appropriately it is thought that a model of the other’s mind is necessary. We investigated this topic further by asking the example questions regarding short passages from Kaland et al. (2005), and the model answered them all correctly. The example passage is given in Table 3.

| **Theory of Mind Example Passage** |
|---|
| Today James is going to Claire’s house for the first time. He is going over for tea, and he is looking forward to seeing Claire’s dog, which she talks about all time. James likes dogs very much. When James arrives at Claire’s house Claire runs to open the door, and her dog jumps up to greet James. Claire’s dog is huge; it’s almost as big as James! When James sees Claire’s huge dog he says,“Claire, you haven’t got a dog at all. You’ve got an elephant! |

Table 3: Example passage from Kaland et al. (2005).

We posed follow-on questions for this passage:

**Q. Is it true, what James says?**

**A.** *No, it’s not true. Claire does have a dog.*

**Q. Why does James say this?**

**A.** *James is surprised because Claire’s dog is so big.*

In a child, this would be considered evidence that the child possesses a well-developed theory of mind (Kaland et al., 2005).

**Counterfactual Knowledge** Some of McCarthy’s questions were about what might have happened, rather than what actually happened. In general, we found GDIB surprisingly good at answering many counterfactual questions. Counterfactual questions requiring temporal or deductive reasoning, however, were often incorrectly answered. (e.g., ***What would have happened if Mr.***

[3] The challenge set questions are in bold italics. The returned answers follow in italics, not bolded.

[4] Incorrect answers are preceded by **.

***Hug had been on the second floor instead of the first?*** *** The car would have crushed him.*)

Counterfactual questions with impossible fictional premises were still usually answered correctly. *(e.g.,* ***What if the salesman had been Superman?*** *He would have been able to fly out of the shaft.*) These kinds of questions create problems for traditional knowledge bases because a consistent knowledge base cannot contain at the same time the set of facts that "no person can fly" and "Superman is a person" and "Superman can fly." GPT has no mechanisms to enforce consistency, which avoids this problem, but at the cost of allowing it to contradict itself and being incapable of guaranteeing anything about the truth of its output.

### 4.2.3 Reasoning

The most challenging questions we posed were questions requiring some kind of reasoning process to arrive at the answer. There has been some success at getting GPT to correctly follow a reasoning process by giving examples of the reasoning steps to follow, and having it imitate these steps one at a time. In the zero-shot prompts we are using, however, reasoning beyond what was required for the earlier types of questions seemed to be beyond its capabilities. It is unclear to what extent these difficulties with reasoning lie with the architecture (a limited number of layers can only carry out so many steps) or with the training set. Certainly other transformers trained on, for example, calculus problems (Lample and Charton, 2019) rather than web text, are able to correctly generate valid chains of reasoning.

**Mathematical Word Problems** Questions that require mathematical operations were frequently incorrect. This matches what one would expect from the original paper on GPT-3, where zero-shot math questions were usually incorrect. (e.g., ***The events of this story happened in 1975. How old would Mr. Hug be in 2020?*** *** He would be 91 years old.*)

GPT-3's abilities at arithmetic and "word problems" have been the subject of several investigations (Gwern, 2020). It is clear that the tokenization makes arithmetic more difficult for the model to learn.

**Temporal Reasoning** GDIB is able to successfully answer questions about what events happened during a particular time interval (e.g., *What happened after the robbers arrived and before they pushed Mr. Hug down the elevator shaft?*) Questions about a time interval that require reasoning about beginnings and ends of events, however, were difficult for the model. These questions were inspired by Kelly and Khemlani (2020). (e.g., ***Did the elevator car reach the springs before Mr. Hug finished falling?*** *** Yes.*)

**False Premises** Questions with false premises were almost never answered correctly. Answering these questions correctly would mean pointing out the error in the question. Instead, the model answers as if the premise of the question were true in a plausible way. (e.g., ***Why was there an airplane in the furniture store?*** *** The airplane was a display in the store.*)

**Insufficient Information** Similarly, questions for which there is not enough information to answer were almost always answered confidently with a plausible (but frequently incorrect) answer (e.g., ***Why is the furniture chain called "Seaman's?"*** *** The chain was founded by a sea captain.*)

In the tests of modified prompts, hinting that "not enough information" is an acceptable answer causes it to get all of the false premise and insufficient information questions right, but causes it to be unwilling to speculate on most of the counterfactual questions, and sometimes to claim there is not enough information when, in fact, there is.

This brings up the important point that answers in this paper provide only a lower bound on what the language model is capable of. It is possible that with a more cleverly designed prompt, the model would be capable of getting more of these questions right. By sticking to a format of story followed by a question and answer prefixed with Q. and A., we have reduced the space of prompts to explore, at the cost of possibly underestimating the full capability of the models.

**Logical Negation** This was a serious problem for smaller models, and still gives GDIB some difficulty. Considering some variations on the first question:

- ***Who was in the store when the events began?*** *John J. Hug*
- ***Who was not in the store when the events began?*** *** The third robber.*
- ***Who was in the store during the events?*** *John J. Hug, the two robbers, and a third robber.*
- ***Who was not in the store during the events?*** *** The robber who attempted to crush Mr. Hug with the elevator car.*

- ***Who was in the store after the events?*** *?? A porter.*[5]
- ***Who was not in the store after the events?*** *?? The robber who attempted to crush Mr. Hug with the elevator car.*

Depending on how one takes "the events" different judgements of these answers are possible, but it is clear that in any case, all the robbers were in the store during the events, and none of them were there after the events. A knowledge base could not contradict itself in this way.

## 5 Analysis

In a zero-shot setting, GDIB seems frequently unable to handle two main types of questions:

- Trick questions: questions with false premises, patterns that do not continue, or not enough information to answer
- Questions that require multi-step reasoning, including mathematical operations (story problems) and anything but the simplest temporal reasoning

A model that would do a better job on questions with not enough information may simply require finetuning on such questions and the appropriate answers, since they occur only relatively rarely in the data GDIB was trained on. (The question about the robbers' names—***What are the names and addresses of the robbers?*** *The names are unknown. The addresses are unknown*—suggests that the model has an appropriately set threshhold for assessing and then describing its uncertainty.) Fixing the problems with reasoning, on the other hand, probably requires a much larger model or changes to the underlying architecture.

On the positive side, GDIB frequently was able both to find and apply relevant background knowledge of the world to correctly answer questions. "What crimes were committed?", for example, seems to require knowing that pushing someone down an elevator shaft is a crime, that it is an attempt to kill them, and that the crime is known by the name "attempted murder." "Who had the money at the end?" seems to require keeping track of the motion of the money through the story. It seems able to predict thoughts and feelings, and to anticipate the immediate result of physical actions.

Beyond supplying answers to questions, we found this striking case where GPT generated this summary text:

***Summarize the story****. A furniture salesman was pushed down the shaft of a freight elevator by two robbers and a third attempted to crush him with the elevator car because they were dissatisfied with the $1,200 they had forced him to give them.*

We also found parameter settings that led the model to change major aspects of the story (this was generated using the stock Davinci model with temperature = 0.7):

***Write a similar story that takes place in ancient Egypt.*** *A wealthy merchant was passing through the Valley of the Kings when a band of marauders descended on him. The merchant was carrying a large amount of gold and jewelry, so the marauders forced him to surrender it. But they wanted more, so they forced him to reveal where he had hidden his most valuable possessions. When he refused to tell them, they put him in a tomb and left him there to die.*

This is a fairly sophisticated analogy. Instead of an elevator shaft, there is a tomb; instead of a furniture salesman with $1200 in Brooklyn there is a merchant with gold and jewels in the Valley of the Kings; but the bones of the story are the same.

One issue with the use of this article as part of a challenge set for language models is that the article may have been part of the training material for GPT, letting it memorize the answers, as discussed in (Lewis et al., 2020) Three lines of evidence show this is unlikely:

1. OpenAI (2019) states that they only trained on pages with outbound links from reddit, to ensure human readability. Searches of reddit.com and search engines indexing it did not turn up any links to any of the versions of the paper found on Google.
2. Performance on our new questions is similar to performance on McCarthy's questions. If his answers were memorized, these would be expected to do worse.
3. None of the answers longer than two words were verbatim, which one would expect if they had been memorized, especially at temperature 0.

Nevertheless, we can't completely rule it out.

## 6 Related Work

**Challenge Datasets** We have constructed a "challenge dataset" guided by approaches in the fields of linguistics and computational linguistics. In the

[5] Answers not clearly correct or incorrect begin with "??".

former, datasets are developed for structured elicitation where linguists ask native speakers to judge grammaticality of presented sentences or to answer questions about their language. The elicitation results are key to deriving properties of the language (Clark et al., 2008; Probst and Levin, 2002). In the latter, datasets are constructed for system evaluation to establish benchmarks to gauge progress on shared tasks. In MT research, going beyond automated scores, the challenge set approach to system evaluation has provided a more fine-grained picture of the strengths of neural systems, as well as insight into which linguistic phenomena remain out of reach (Isabelle et al., 2017). In recent LM research that probes the syntactic knowledge in these models, the structured datasets of minimal grammatical/ungrammatical sentence pairs test when LMs assign a higher probability to the grammatical sentence than the ungrammatical one (Marvin and Linzen, 2018; Newman et al., 2021). Though significant caution is needed in interpreting LM results, we see diagnostic value in constructing challenge datasets to evaluate systems' *passage understanding*, as carefully constructed by Dua et al. (2019). The fact that a current state-of-the-art LM dropped more than 50 absolute F1 points on their dataset reinforces our belief that challenge datasets can spur research into more comprehensive semantic analyses of what LMs know, encouraging system hill-climbing up the right hill, per Bender and Koller (2020).

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is of particular relevance to this work, as it is a collection of resources for both training and evaluation of various types of Natural Language Understanding tasks. It is intended to be agnostic to the system type. The evaluation suite includes tasks related to sentiment, paraphrase, natural language inference, coreference, as well as question-answering. While we took inspiration from this benchmark in our inclusion of questions overlapping with the inference tasks in GLUE, such as the Winograd Schema Challenge, we contribute a set of challenge questions that have been carefully crafted and classified to probe the strengths and weaknesses of a LM with respect to particular knowledge sources, types, and reasoning requirements.

**LMs and "prompt-based learning"** Numerous research teams are working to evaluate what LMs may "know", even asking how they compare to knowledge bases (Petroni et al., 2019). New possibilities are emerging with methods of estimating the knowledge in LMs that can be found with automatically constructed prompts that yield better results than those manually created, demonstrating that any given prompt may be sub-optimal (Jiang et al., 2020). Given our focus on assessing what one family of prompt-based LMs can answer about a given text passage, the most recent and pertinent overview that situates these LMs in the field is the extensive, systematic survey of prompting methods and pre-trained language models using these methods by Liu et al. (2021). As this survey notes, in tuning-free prompting as we have carried out with GPT LMs, the questions in the q/a task generate the answers directly. This is efficient, yet also leaves the prompts as the only method to provide the task specification. For this reason, we present the framework of example questions that serve as test prompts in a structured challenge dataset both so that they can be reused by others in testing their systems, and so that we can extend these over time.

## 7 Conclusion

This limited evidence suggests that current language models are able to supply answers to questions about a passage in a way that, at least superficially, meets the challenge of "natural language understanding" as originally set out in the 1970s. They can do surprisingly well at describing the likely thoughts and feelings of characters though these are not explicit in the passage. The models can also generate text that describes likely consequences for what might have happened if things had gone differently, and that fills in the factual information that we tend to omit when using natural language. However, many questions requiring even a little careful thought, reasoning, or multi-step inference go beyond the capability of these models.

For those who want to make use of these models today, this suggests sticking to applications where the information needed to answer the questions are immediately available in the passage or slowly-changing, widely known information about the world; or else creative questions where there is no wrong answer. For researchers, though, this highlights the need to discover ways of combining the deductive reasoning capabilities already present in early AI work with the context-sensitivity and ability to work with natural language that these models provide.

## References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jonathan Clark, Robert E. Frederking, and Lori S. Levin. 2008. Toward active learning in data selection: Automatic discovery of language features during elicitation. In *LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *2019 NAACL: Human Language Technologies, Volume 1*, Minneapolis, Minnesota. Association for Computational Linguistics.

Leo Gao. 2021. On the sizes of openai api models.

Gwern. 2020. Gpt-3 creative fiction.

Pierre Isabelle, Colin Cherry, and George F. Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2486–2496. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nils Kaland, Annette Møller-Nielsen, Lars Smith, Erik Lykke Mortensen, Kirsten Callesen, and Dorte Gottlieb. 2005. The strange stories test. *European child & adolescent psychiatry*, 14(2):73–82.

Laura Jane Kelly and Sangeet Khemlani. 2020. Directional biases in durative inference. In *CogSci*.

Jonathan L Kvanvig. 2015. Knowledge, understanding, and reasons for belief. In *The Oxford Handbook of Reasons and Normativity*.

Guillaume Lample and François Charton. 2019. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*.

Belinda Z Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

John McCarthy. 1990. An example for natural language understanding and the ai problems it raises. *Formalizing Common Sense: Papers by John McCarthy*, 355.

Erik T Mueller. 1999. Prospects for in-depth story understanding by computer.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining targeted syntactic evaluation of language models. *CoRR*, abs/2104.09635.

OpenAI. 2019. Better language models and their implications.

F. Petroni, T. Rocktäschel, A. H. Miller, P. Lewis, A. Bakhtin, Y. Wu, and S. Riedel. 2019. Language models as knowledge bases? In *In: EMNLP 2019*.

Katharina Probst and Lori Levin. 2002. Challenges in automated elicitation of a controlled bilingual corpus. In *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-02)*.

Walid Saba. 2020. Language and its commonsense: Where formal semantics went wrong, and where it can (and should) go. *Journal of Knowledge Structures and Systems*, 1(1).

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450.