Are GPT-3 Models Pragmatic Reasoners?

Stanford CS224N Custom Project

Ariane Lee
Department of Computer Science
Stanford University
`arialee@stanford.edu`

Abstract

While large language models (LLMs) have shown incredible performances over natural language understanding (NLU) tasks, a question that remains to be solved is whether they are pragmatic listeners that can reason over ambiguous scenarios and act optimally. In this project, we test the pragmatic reasoning capabilities of GPT-3 models in ambiguous scenarios under the Rational Speech Act (RSA) framework in a reference game setting. Testing the models with zero-shot, few-shot, and few-shot chain-of-thought (CoT) prompting, we find that although few-shot prompting produces little performance improvement, and indeed often worsens model performance, few-shot CoT prompting dramatically improves the performance of davinci models.

1 Key Information to include

- Mentor: Zhengxuan Wu- `wuzhengx@stanford.edu`
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Human communication is largely affected by its context. When we speak and when we listen, our utterances as speakers and our interpretations as listeners are inevitably sensitive to the expectations of the others involved in our communication and the context of the communication. For example, if you asked someone "Do you want to grab dinner?", and they say "I have work to do.", without them explicitly saying no, you know that the answer is implicitly no; this is because we use a skill called pragmatic reasoning, where we can glean information from context. The example above is an implicature, where a person means to convey something that is not explicitly said (Maru and Bevilacqua, 2021).

In the era of ChatGPT and models from the GPT family, large language models (LLMs) exhibit astonishing performance over NLU tasks, and start to serve as daily agents that help humans to accomplish various tasks. They are clearly literal listeners in the sense that they can understand human questions or conversations and generate utterances that communicate faithfully. One big question that remains to be solved is whether they are pragmatic listeners that can reason over ambiguous scenarios and act optimally. Among other intricacies of human language such as idioms, ambiguous scenarios, like the one depicted in Figure 1, are difficult for computers to model and understand. While these ambiguous scenarios are similar to implicatures, the presence of a misleading choice requires additional reasoning for one to arrive at the correct conclusion. In the example, the listener has to reason about the speaker's specific choice of choosing "glasses" to describe her friend, noting although two options wear glasses, had the speaker intended to refer to the option on the right, it would have been clearer to describe her friend as wearing a hat. Thus, the middle option is the option the speaker refers to.

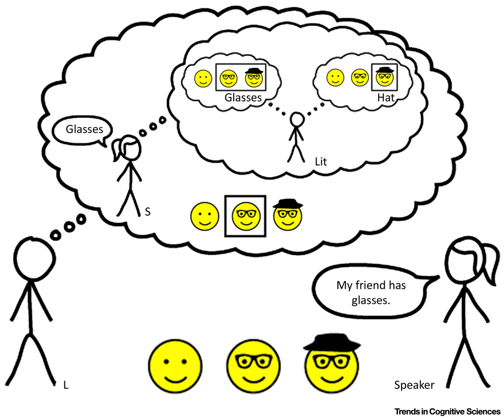

Figure 1: Ambiguous scenario illustration adapted from Goodman and Frank (2017)

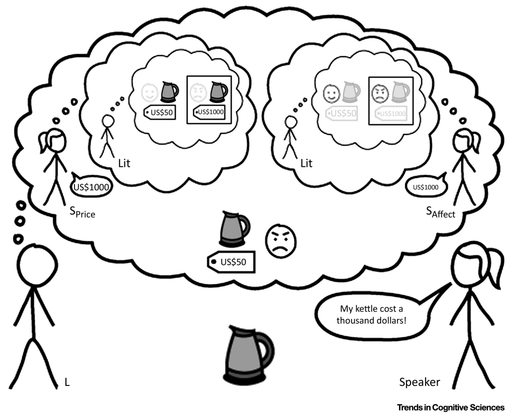

Figure 2: Hyperbole reasoning illustration adapted from Goodman and Frank (2017)

We test the pragmatic reasoning abilities of GPT-3 models under the well-studied Rational Speech Act (RSA) framework in a reference gaming setting, specifically using ambiguous list comprehension prompts. What distinguishes our work is the application of RSA to to these ambiguous scenarios.

As human communication frequently invokes the use of pragmatic reasoning to deduce understandings that cannot be gleaned from just literal interpretations. This can occur in the form of literal ambiguous scenarios with choices, like scheduling, for example, or more implicit ambiguous scenarios that appear in human language like hyperboles (Figure 2). For LLMs to more effectively evaluate and understand human communication, they have to be able to pragmatic reasoning themselves, a development that would have a wide variety of applications among conversational agents.

3 Related Work

Monroe et al. (2017) conducted research that is analogous to this project in the sense that it similarly focuses on utilizing the RSA framework for its models. The main differentiators are the models being used (their own novel models vs GPT-3 models) and the specifics of the tasks (visual and language based vs just language based); their tasks focus on ambiguous color references.

Ruis et al. (2023) investigates whether LLMs have the ability to make this type of implicature inferences, finding that despite only evaluating on utterances that require a binary inference (yes or no), most perform close to random. While this work similar establishes a baseline for evaluating language in context, while they focus on implicatures, we will have models evaluate explicitly ambiguous prompts.

4 Approach

4.1 Task

We focus on applying the RSA framework in a reference gaming setting, focusing on tasks that have speaker and listener agents. After creating our datasets, as our initial task, we test to see if the GPT-3 and GPT-.5 models understand and respond accurately to non-ambiguous tasks. Specifically, we start with literal list comprehensions. For example, given the scenario in Figure 3 where a speaker presents several lists in a prompt, can the model, acting as the listener, identify which list the speaker is referring to when the answer is non-ambiguous? We then see if the GPT-3 models can accurately reason through similar pragmatic list comprehensions where the answer is ambiguous via recursive reasoning about the speaker's intentions. We then experiment with providing varying amounts of in-context examples of pragmatic list comprehension tasks (few-shot prompting) to see if they evoke any improvement in reasoning ability. Following experimenting with few-shot prompting, we include answer rationales in our in-context examples, chain-of-thought (CoT) reasoning in an effort to elicit better model performance. Across all of these tasks, we evaluate individual models' performances to see if successive generations of GPT are better at pragmatic reasoning.

> Task Description
This is a reference game, where **there are one speaker and one listener**. Without giving the listener the true answer, the speaker's job is to tell the listener to pick the object that the speaker has in mind. **Whenever there are ambiguous answers, the listener need to reason out the answer by picking the best one.** You are acting as the listener.

Context: List A = [1,2], List B = [3,4], List C = [5]
Speaker: Pick the List that contains 3.
Listener (You): The answer is B.

[More examples abbreviated due to length limit]

Context: List A = [9,11], List B = [12], List C = [3,14]
Speaker: Pick the List that contains 9.
Listener (You): The answer is A.

★ Reference target is not ambiguous.
★ (Or) There is **no ambiguous distractor.**

Figure 3: Example of ambiguous pragmatic reasoning task

4.2 Methods

To the best of our knowledge, there are currently no datasets that exist for us to test pragmatic reasoning of ambiguous tasks under the RSA framework. The most similar benchmark that was found was created by Auther and Author, but it tests for understanding of conversational implicatures, which are similar but differ in complexity to the ambiguous scenarios we propose. Thus, we have constructed our own datasets. Our datasets contain list comprehension prompts within the RSA framework, and they vary in terms of ambiguity, number of in-context examples, and whether or not these examples include CoT reasoning. The dataset construction process is further elaborated upon in Section 5.1.

4.3 Baselines

Since there is no currently existing benchmarks on the specific task we will be focusing on, our baseline will be the models' performance on non-ambiguous prompts. A model's performance on non-ambiguous prompts will indicate whether it can faithfully understand our prompt and act as literal listeners.

4.4 Models

The models we plan to use are various generations of GPT-3 models. More specifically, we will be using text-davinci-003, text-davinci-002, text-curie-001, text-babbage-001, and text-ada-001 The use of all 5 models will allow us to test the emergence abilities of LLMs.

5 Experiments

5.1 Dataset Types

We constructed 6 different datasets: non-ambiguous, ambiguous 0-shot, ambiguous 5-shot, ambiguous 10-shot, ambiguous 5-shot with CoT reasoning, and ambiguous 10-shot with CoT reasoning. Each dataset consists of 1000 prompts. Every prompt is a list comprehension problem that is presented with a set of instructions (Figure 4) that set the RSA framework. Each prompt has 3 list choices, two list choices with two numbers, and one list choice with one number. A minimum of 1 and maximum of 100 were set to determine the range of these numbers. The numbers for the lists were chosen by randomly sampling without replacement from this range using random.sample().

Parameters There are 3 parameters that distinguish our datasets' prompts from each other: ambiguity, number of in-context examples, and whether or not these examples include rationales. All of the datasets have prompts with ambiguous answers except one, a non-ambiguous prompt data set which is used to test the baseline literal listener reasoning abilities of the models. A non-ambiguous prompt (Figure 5) has list choices that are clearly distinct, with only one list choice containing the target number mentioned in the speaker's instructions, meaning there one obvious answer. Similar to a non-ambiguous prompt, an ambiguous prompt (Figure 6) has 3 list choices, but two of them contain the target number mentioned by the speaker, creating ambiguity. The way to arrive at the target list choice is to evaluate the reasoning behind the speaker's specific choice of target number (Figure 6), which is how the RSA framework is utilized within the context of our ambiguous prompts.

For the datasets that include in-context examples (ambiguous 5-shot, ambiguous 10-shot, ambiguous 5-shot with CoT reasoning, and ambiguous 10-shot with CoT reasoning), the examples closely mimic the structure of the prompts except that the answer is provided. Where a prompt would end with "The answer is List", an example would be completed with "The answer is List A." For the datasets that include rationales with their examples, the rationales precede the ending of "The answer is List...".

Instructions:
This is a reference game, where there is one speaker and one listener. Without giving the listener the true answer, the job of the speaker is to tell the listener to pick the object that the speaker has in mind. Whenever there are ambiguous answers, the listener needs to reason out the answer by picking the best one. You are acting as the listener

Figure 4: Instructions attached to each prompt

Non-ambiguous prompt:
Context: List A =[34], List B = [67, 51], List C = [43, 83]
Speaker: Pick the list that contains 67.
Listener (You): The answer is List

Figure 5: Example of non-ambiguous prompt

Ambiguous prompt:
Context: List A =[98, 41], List B = [50, 98], List C = [41]
Speaker: Pick the list that contains 98.
Listener (You): Let's think step by step. Both List A and List B contain 98, resulting in an ambiguous answer. To pick the best answer, let's assume that the speaker wants the listener to pick List B. If so, the optimal way to convey this message is asking the listener to pick the list that contains 50, since only List B contains 50. But the speaker chooses not to say this. Thus, the answer is not List B. The answer is List A.

Figure 6: Example of ambiguous example with CoT reasoning

5.2 Prompt Engineering

There were several ways that we purposefully engineered the prompts in our datasets.

Ordering Each of our list comprehension problems has 3 choices: the target answer, a misleading choice that technically could be correct due to ambiguity, and an obviously incorrect choice. If we were to present prompts or examples where the 3 choices appeared in the same order each time, we were concerned that the GPT-3 models would attach a relevance to the sequencing of the choices, and choose answers off of that. Thus, we were cautious to avoid that potential scenario by randomizing the orders of the choices. Since there are 6 possible orderings for these 3 choices (3!), this was accomplished by randomly generating an integer from 1-6 using random.randint() each time a prompt or example was generated to determine the ordering.

Open-Endedness One other concern of ours was the lack of regularity that could come with prompts that were too open ended. For example, after being instructed to pick a list containing a certain number, there are a number of responses that the model could generate. We were especially cautious about avoiding a scenario where a model would answer as the listener by returning not just one, but two lists, effectively avoiding reasoning and selecting both ambiguous answers (i.e. returning "List A and List B"). To combat this issue, we ensured that the last portion of each prompt leading up to the model generated answer was "The answer is List", with the is implying only one choice should be returned as the answer (The choices being "A", "B", and "C"). For ease of answer detection, we also ensured that "List" was the final word of each prompt so that the model would generate a response that was a letter that would refer to the list name rather than the list itself ("A" vs "[1, 3]"). The exception to this was generated answers to prompts that included rationales. In those cases, generated answers followed the structure of the rationale, but typically ended in the same sentence as the generated answers to prompts without rationales.

Rationale To elicit reasoning, each rationale starts with "Let us think step by step", a phrase shown to have improved CoT by **?**. The rationale then explains the correctness of a choice by exploring the speaker's intentions, providing reasoning for the specific selection of given target number and illustrating why that specific selection implies a certain choice, before stating that the misleading ambiguous choice is incorrect and that the target choice is the answer.

5.3 Evaluation method

We have casted our problem as a classification task, for which we report on precision, recall, accuracy,and F1 scores. There are currently no existing benchmarks regarding the specific task of reasoning through written prompts within a RSA framework. Monroe et al.'s (2017) Colors in Context: A Pragmatic Neural Model for Grounded Language Understanding can be seen as analogous to this study, but a direct comparison cannot be made due to the inherently different natures of the tasks (visual vs. linguistic). Thus, we rely on the baseline of random accuracy to judge the accuracy or our models. In an unambiguous case, the baseline random accuracy will be $1/n$, where n is the number of choices, as we expect the model to return one token as its response. We originally anticipated having to define a metric to incorporate situations where the model returns more tokens than expected in an ambiguous case. For example, if there are three choices, A, B, and C, and A B are the ambiguous choices, the model could return "A or B." However, we avoided this complication by adjusting our prompts to ensure only one choice was returned each time. This method was effective, with only 5 extraneous answers from a total of 30,000 generated answers.

5.4 Experimental details

Across all models, we set the parameters to be the same. Temperature was set to 0, as classification tasks should have no ambiguity, maximum length was set to the default of 256, as each response should be quite short, and Top P was set to 1.

5.5 Results

The quantitative accuracy measures of the models' performances on the datasets are pictured in the graphs below (Figures 7-12). A dotted green line indicating baseline measure of random accuracy was included for understanding. To see all of the metrics in tabular form, Figures (15-20) in the appendix.

Even on non-ambiguous prompts with a clear answer, text-ada-001, text-babbage-001, and text-curie-001, henceforth referred to as older GPT-3 models, performed poorly. Improvement remained poor for the older models across all datasets. This is exemplified by the oldest model, text-ada-001, which achieves around .333 accuracy for all of the datasets. The text-davinci-002 and text-davinci-003 models, henceforth referred to as the newer models, however, achieved 100% accuracy on the non-ambiguous dataset.

All models performed poorly on the ambiguous 0-shot dataset, although the newer models tend to do slightly better. As in-context examples are added into ambiguous prompts, models do not perform significantly better; at times they perform worse, as shown by the decreases in accuracy from text-curie-001, text-davinci-002, and text-davinci-003 when going from evaluating on the ambiguous 0-shot dataset (Figure 8) to the ambiguous 5-shot dataset (Figure 9).

For the older models, there were no significant increases in performance when going from evaluating on ambiguous prompts with examples (Figures 9 and 10) to evaluating on ambiguous prompts with examples are provided with CoT reasoning (Figures 11 12). However, newer models perform much better on datasets where prompts are given examples with CoT reasoning (Figure 11), and this performance improvement increases with additional provided examples with CoT reasoning (Figure 12).

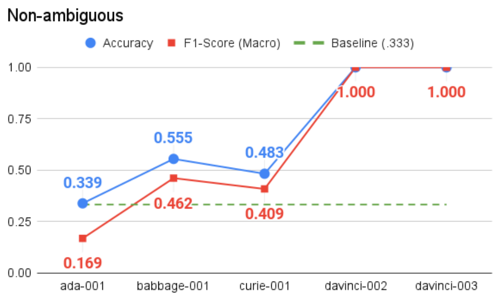

Figure 7: Non-ambiguous dataset results

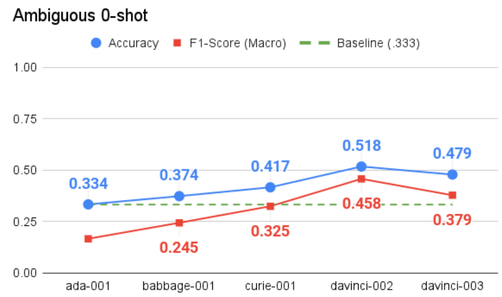

Figure 8: Ambiguous 0-shot dataset results

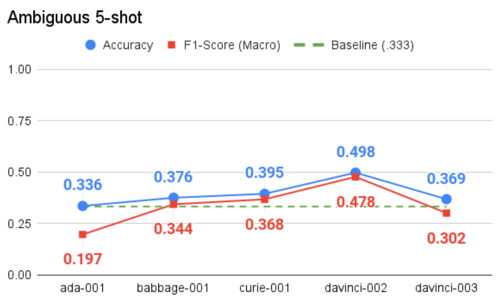

Figure 9: Ambiguous 5-shot dataset results

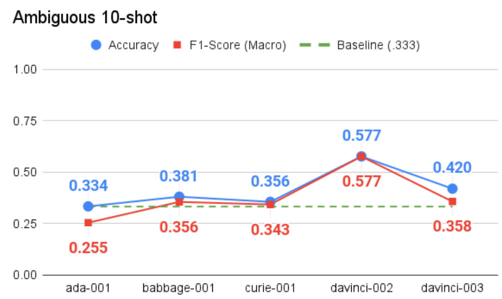

Figure 10: Ambiguous 10-shot dataset results

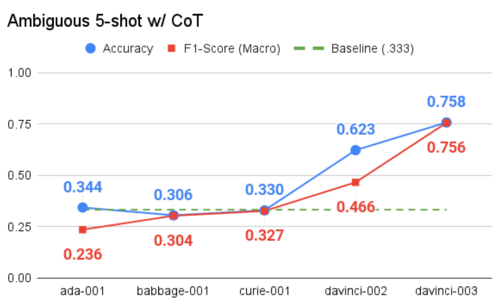

Figure 11: Ambiguous 5-shot w/ CoT reasoning dataset results

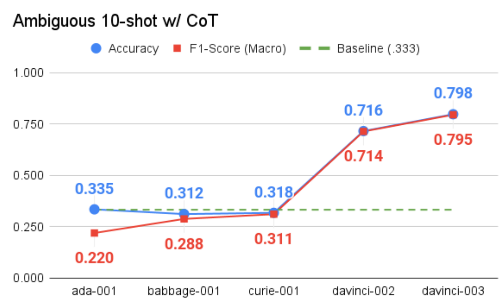

Figure 12: Ambiguous 10-shot w/ CoT reasoning dataset results

While there were still generated answers that returned more than the desired number of tokens, across all of the generated answers (total 30,000 answers), only 5 generated answers fell outside of this norm, and the maximum number of extraneous generated answers from one dataset was 2. Four of the five extraneous generated answers were produced by text-ada-001, while 1 was produced by text-davinci-002.

Example of extraneous answer

6 Analysis

Understanding of non-ambiguous prompts From the results, it seems like older models cannot even be considered literal listeners. Even when presented with a prompt with one clear answer, they barely perform better than baseline random accuracy. Closer looks at the generated answers showed that all of the older models had a baseline tendency towards generating the answer of "C." In fact, text-ada-001, text-babbage-001, and text-curie-001 generated almost all "C"'s in response to the non-ambiguous and ambiguous 0-shot datasets, indicating that they required in-context examples to even really start considering other potential answers. The older models start producing a slightly more diverse answers across once examples are provided in-context, but a bias towards "C" is still apparent.

Lack of performance improvement with few-shot None of the models showed significant performance improvements with added in-context examples, except for a slight uptick in text-davinci-002's performance when going from ambiguous 5-shot prompts to ambiguous 10-shot prompts. This is distinct from results of past work that show performance improvements with added examples. This leads to the conclusion that providing in-context examples is not enough to significantly improve model performance when it comes to ambiguous prompts.

Understanding of CoT reasoning The older models start producing a much more even set of answers across "A", "B", and "C" once rationales are provided with the examples, but this doesn't improve their accuracy. This indicates that the older models don't really understand the CoT reasoning to the point of being able to reach the right answer, and that they are likely just imitating variances in the correct answers in the examples or another non-apparent pattern. The lack of understanding of the rationales is apparent when looking at individual examples of generated answers.

A prompt and answer generated by text-curie-001 that the model got wrong is depicted in Figure 13. Not only is the model unable to detect which lists actually share the target number, it also incorrectly states that List B has a unique number of 27. The model then gives an answer of List A, which wasn't even mentioned in any of the prior reasoning steps. Answers with the same errors were produced from text-ada-001 and text-babbage-001.

Prompt:
Context: List A = [57], List B = [22, 12], List C = [12, 57]
Speaker: Pick the list that contains 11
Listener (You): Let us think step by step.

text-curie-001 Generated Answer:
Both List A and List C contain 12, resulting in an ambiguous answer. To pick the best answer, let us assume that the speaker is listening to pick List C. If so, the optimal way to convey this message is asking the listener to pick the list that contains 57, since only List C contains 57. But the speaker chooses not to say this. Thus, the answer is not List C. The answer is List B.

text-davinci-002 Generated Answer:
Both List B and List C contain 12, resulting in an ambiguous answer. To pick the best answer, let us assume that the speaker is listening to pick List C. If so, the optimal way to convey this message is asking the listener to pick the list that contains 57, since only List C contains 57. But the speaker chooses not to say this. Thus, the answer is not List C. The answer is List B.

Target Answer:
List C

Figure 13: Prompt that both text-curie-001 and text-davinci-002 classified incorrectly

Figure 14 shows a prompt and answer generated by text-curie-001 that the model got right, however, despite producing the correct list answer, the rationale provided by the model clearly indicates that it did not utilize CoT reasoning to reach the answer, and likely produced the right answer by chance.

On the other hand, text-davinci-002 and text-davinci-003 indicate a better understanding of CoT. When producing answers, most correct answers follow CoT reasoning with sound logic, such as depicted in text-davinci-002's generated asnwer in Figure 14. Even when these models product incorrect answers, the errors in their rationales are much less significant that the ones in the older models' incorrect rationales. In Figure 13, we see that although text-davinci-002 produced an incorrect answer, the model was still able to correctly identify which two lists had the target number in the first line of reasoning. In addition, unlike the final incorrect answer generated in Figure 13 by text-curie-001, the final incorrect answer generated by text-davinci-002 is at least mentioned in earlier reasoning steps.

Prompt:
Context: List A = [18, 17], List B = [17, 59], List C = [59]
Speaker: Pick the list that contains 17
Listener (You): Let us think step by step

text-curie-001 Generated Answer:
Both List A and List C contain 17, resulting in an ambiguous answer. To pick the best answer, let us assume that the speaker is listening to pick List C. If so, the optimal way to convey this message is asking the listener to pick the list that contains 59since only List C contains 59. But the speaker chooses not to say this. Thus, the answer is not List C. The answer is List B.

text-davinci-002 Generated Answer:
Both List A and List B contain 17, resulting in an ambiguous answer. To pick the best answer, let us assume that the speaker is listening to pick List A. If so, the optimal way to convey this message is asking the listener to pick the list that contains 18since only List A contains 18. But the speaker chooses not to say this. Thus, the answer is not List A. The answer is List B.

Target Answer:
List B

Figure 14: Prompt that both text-curie-001 and text-davinci-002 classified correctly

7 Conclusion

In this project, we built a benchmark for evaluating pragmatic reasoning in ambiguous scenarios under the RSA framework, and we tested GPT-3 models on our benchmark. We find that the baseline performances of the models on ambiguous 0-shot prompts are poor. When evaluating on ambiguous scenarios, models do not perform better with in-context examples alone. It is only when CoT reasoning is provided with examples that any model's performance improves significantly; this happens with text-davinci-002 and text-davinci-003. In light of these results, it would be interesting to further investigate if models can understand ambiguous prompts in more day to day contexts, for example, with meeting scheduling instead of list comprehensions. Several papers have also recently been released regarding multimodal CoT reasoning (Zhang et al., 2023) (Huang et al., 2023). Running experiments with multimodal CoT reasoning with ambiguous multimodal prompts would be antoher exciting avenue to explore. It would also be interesting to explore experiment with more in-depth, descriptive CoT reasoning to see if that elicits better results or to have our datasets evaluated by humans for the construction fo a human baseline.

References

Noah Goodman and Michael Frank. 2017. Pragmatic language interpretation as probabilistic inference. In *Trends in Cognitive Sciences*, pages 818–829, Online.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Lv Tengchao, Cui Lei, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav. Chaudhary, Som Subhojit, Xia Song, and Furu Wei. 2023. Language is not all you need: Aligning perception with language models. Online.

Marco Maru and Michele Bevilacqua. 2021. Beyond the imitation game benchmark. Online.

Will Monroe, Robert Hawkins, Noah Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. In *Transactions of the Association for Computational Linguistics*, pages 325–338, Online.

Laura Ruis, Akbir Khan, Stella Biderman, Hooker Sara, Tim Rocktäschel, and Edward Grefenstette. 2023. Pragmatic language interpretation as probabilistic inference. Online.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. Online.

A Appendix (optional)

Non-Ambiguous	Trial 1: # Correct	Trial 2: # Correct	Total # Correct	Total Trials	Accuracy
text-ada-001	172	167	339	1000	0.339
text-babbage-001	286	269	555	1000	0.555
text-curie-001	243	240	483	1000	0.483
text-davinci-002	500	500	1000	1000	1
text-davinci-003	500	500	1000	1000	1

Figure 15: Non-ambiguous dataset results

Ambiguous 0-shot	Trial 1: # Correct	Trial 2: # Correct	Total # Correct	Total Trials	Accuracy
text-ada-001	151	183	334	1000	0.334
text-babbage-001	175	199	374	1000	0.374
text-curie-001	199	218	417	1000	0.417
text-davinci-002	276	242	518	1000	0.518
text-davinci-003	257	240	497	1000	0.497

Figure 16: Ambiguous 0-shot dataset results

Ambiguous 5-shot	Trial 1: # Correct	Trial 2: # Correct	Total # Correct	Total Trials	Accuracy
text-ada-001	159	177	336	1000	0.336
text-babbage-001	188	188	376	1000	0.376
text-curie-001	202	193	395	1000	0.395
text-davinci-002	254	244	498	1000	0.498
text-davinci-003	200	172	372	1000	0.372

Figure 17: Ambiguous 5-shot dataset results

Ambiguous 10-shot	Trial 1: # Correct	Trial 2: # Correct	Total # Correct	Total Trials	Accuracy
text-ada-001	163	171	334	1000	0.334
text-babbage-001	195	186	381	1000	0.381
text-curie-001	178	178	356	1000	0.356
text-davinci-002	290	287	577	1000	0.577
text-davinci-003	205	215	420	1000	0.42

Figure 18: Ambiguous 10-shot dataset results

Ambiguous 5-shot with Rationale	Trial 1: # Correct	Trial 2: # Correct	Total # Correct	Total Trials	Accuracy
text-ada-001	177	167	344	1000	0.344
text-babbage-001	145	161	306	1000	0.306
text-curie-001	156	174	330	1000	0.33
text-davinci-002	310	315	625	1000	0.625
text-davinci-003	383	375	758	1000	0.758

Figure 19: Ambiguous 5-shot with CoT reasoning dataset results

Ambiguous 10-shot with Rationale	Trial 1: # Correct	Trial 2: # Correct	Total # Correct	Total Trials	Accuracy
text-ada-001	164	171	335	1000	0.335
text-babbage-001	148	164	312	1000	0.312
text-curie-001	150	168	318	1000	0.318
text-davinci-002	352	364	716	1000	0.716
text-davinci-003	401	397	798	1000	0.798

Figure 20: Ambiguous 10-shot with CoT reasoning dataset results