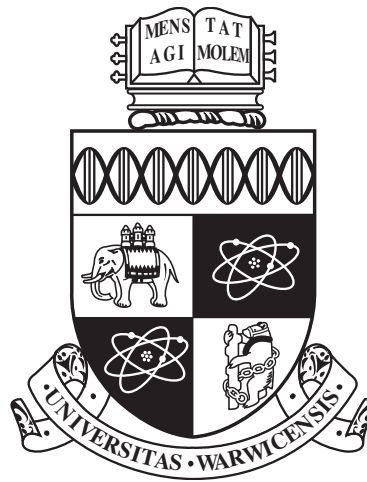


Evaluating Prompt Engineering Strategies to Enhance Theory of Mind in Large Language Models



by

Kailai He

Supervisor: Gabriele Pergola

Department of Computer Science

University of Warwick

April 2024

Abstract

Theory of Mind (ToM) is the ability to attribute mental states such as beliefs, intentions, and emotions to oneself and others, which is crucial for understanding human interactions. Recent advancements in Large Language Models (LLMs) have led to models with increasing linguistic and reasoning capabilities. However, their capacity to perform tasks involving ToM remains limited, as they struggle to accurately infer the beliefs and intentions of others in various contexts. In this dissertation, we systematically investigate the potential of prompt engineering to enhance ToM performance in LLMs.

We explore strategies such as Chain of Thought (CoT) reasoning, Double Inference, and structured JSON prompts to improve the understanding of mental states in models like Flan-T5-base and Zephyr-7b-beta. Evaluating these methods using benchmarks like SocialIQA and ToMi, we analyze how carefully designed prompts impact model reasoning and performance in ToM tasks. The findings of this project reveal that prompt engineering significantly influences the ToM capabilities of LLMs, analysing both the limitations and the potential for improvement in these models.

Keywords: Large Language Models, Theory of Mind, Prompt Engineering, Natural Language Processing.

Acknowledgments

I would like to thank my project supervisor, Dr. Gabriele Pergola. Especially, his guidelines have helped to shape the structure of the project. Without his advice, I would have found navigating both direction and time management more challenging. I would also like to thank my aunt for her advice on the project.

Contents

Abstract	ii
Acknowledgments	iii
List of Figures	iii
List of Tables	1
Chapter 1 Introduction	1
1.1 Prompt Engineering and ToM	2
1.2 Evaluating ToM Abilities of LLMs	2
1.3 Contribution	3
Chapter 2 Background	4
2.1 Theory of Mind	4
2.1.1 Theory of Mind (ToM) and Pragmatic Ability	6
2.1.2 An Alternative Approach to Define ToM Tasks	8
2.2 ToM and LLMs: Have ToM Capabilities Spontaneously Emerged in LLMs?	9
2.2.1 ToM Tasks that LLMs Fail	11
2.3 Prompt-Engineering to Improve ToM Capabilities	13
2.3.1 Step-By-Step Reasoning	13
2.4 ToM Datasets	15
2.4.1 SocialIQA	16
2.4.2 ToMi	18
2.4.3 FanToM	19
Chapter 3 Methodologies	21
3.1 Prompt Engineering for ToM	21
3.1.1 Simple Prompting Strategies for ToM	21

3.1.2	Chain-of-Thought Prompting	22
3.1.3	OverPrompt	22
3.1.4	Structured Prompts - JSON format	24
3.1.5	Double Inference - A Novel Prompt Strategy	25
3.2	ToM Benchmarks: A Detailed Analysis	26
3.3	Parsing LLM Outputs for Evaluation	28
3.3.1	Direct Inference	28
3.3.2	Cosine Similarity	30
Chapter 4	Evaluation	33
4.1	Datasets	34
4.1.1	SocialIQA	34
4.1.2	ToMi	35
4.2	Models	35
4.2.1	Flan-T5-base	35
4.2.2	Llama3-8B	36
4.3	The Mistral Family	37
4.3.1	Mistral-7b	37
4.3.2	zephyr-7b-beta	39
4.4	Experiments	40
4.4.1	Experimental Results	40
4.4.2	Qualitative Analysis	41
Chapter 5	Project Management	44
5.1	Running Time	44
5.2	JSON Format	45
5.3	TimeTable	45
Chapter 6	Conclusion	48
6.1	Contribution	48
6.2	Limitations	48
6.3	Future Work	49
Appendix A	Cosine Similarity Check	50
Appendix B	GPT4 on Altered Tasks	55
Bibliography		59

List of Figures

2.1	An image depiction of the Sally-Ann task. The woman represents Sally, and the man represents Ann. The dotted line box represents a closed room; people outside of the room will not know anything that is happening inside the room. The basket and the box (represented as a gist box) are non-transparent.	6
2.2	Accuracy on the SocialIQA dev. set, broken down by LLM model type and size, as well as number of few-shot examples (k). Figure from [27]	17
2.3	Accuracy on the ToMi dev. set MIND questions of varying sizes of GPT-3 and with varying number of examples (k). Figure from [27].	19
3.1	OverPrompt example. Figure from OverPrompt [17].	23
3.2	Double Inference example: The key is to concatenate the original question, the model output, and the phrase 'Therefore, the answer is' into a single prompt. This concatenated prompt is then inputted back into the model to extract a label from the composite answer.	26
4.1	Example of socialIQA data, bold text is the key, and plain text is value	34
4.2	Example of how socialIQA data is formatted	34
4.3	Example of ToMi data	35
4.4	Example of zephyr output.	42

List of Tables

2.1	Example of a Chain of Thought (CoT) response.	15
2.2	Examples of SocialIQA questions focusing on different persons (Agent, Others). Figure from [27].	17
2.3	Example stories in the ToMi dev data set. "Type" denotes reasoning type, with M-1 and M-2 representing MIND-1st and MIND-2nd, respectively.	18
3.1	Performance Metrics for Different Prompt Sizes	24
4.1	Question and Answer Context.	37
4.2	Accuracy of different Large Language Models across various datasets and prompting strategies. 'PR' indicates 'Phrase Removed,' 'CoT' refers to 'Chain-of-Thought,' and 'D-I' denotes 'Double Inference. . .	40
5.1	Work schedule.	47
5.2	original Term 2 timetable.	47

Chapter 1

Introduction

Artificial Intelligence (AI) has swiftly risen to prominence, captivating interest with its broad capabilities that enhance daily workflows across various fields. Among these, the rise of Natural Language Processing (NLP) technologies, particularly through developments in large language models (LLMs) like ChatGPT ¹, has showcased substantial linguistic prowess and human-like abilities, contributing significantly to AI's popularity.

This growing capability extends beyond mere technical tasks. The human-like faculties of these models, including their ability to engage in complex communication and exhibit understanding similar to human cognitive functions, have attracted attention from various fields. Researchers in Cognitive Sciences have begun to delve into the phenomenon, investigating these models' abilities to mimic human mental processes through the Theory of Mind (ToM). ToM assesses one's capability to attribute mental states—beliefs, desires, intentions—to oneself and others, a fundamental aspect of human social interactions [15] [3] [19]. Despite their impressive advancements, LLMs often struggle with ToM tasks. Our report will focus on various ToM studies conducted on LLMs, exploring their performance, utility, and strategies for enhancing such capabilities. Notably, despite early models showing poor results in ToM-style tests [25, 27], recent advancements with models like GPT-4 and GPT-3.5 have seen remarkable improvements, scoring high in specialized ToM probing datasets like SocialIQA and ToMi [15, 19]. This indicates that as language models grow more complex, their ability to perform ToM-related tasks also improves, either as a direct target of their design or as a by-product of their general capabilities.

¹<https://chat.openai.com/>

1.1 Prompt Engineering and ToM

To assess and improve ToM abilities in LLMs, prompt engineering is a promising technique as it allows the design and structure of prompts that can significantly influence model performance. Prompt engineering is a technique used to design and refine the prompts given to LLMs to enhance their performance on specific tasks. With the increasing sophistication of the models, the way instructions are framed can significantly influence the quality and accuracy of the output. This involves carefully crafting input queries to guide the models in understanding the task, providing context, and leading to more accurate and relevant responses. Therefore, prompt engineering has become crucial in areas like reasoning tasks, where the formulation of a prompt can mean the difference between a correct or incorrect response. In this work, we explore different prompting strategies to address the limitations of LLMs in ToM tasks. These strategies are based on guided reasoning (e.g., Chain-of-Thought[31] (CoT)), grouping requestes (e.g., Overprompt), or structured instructions (e.g., following the JSON format). Along with these existing strategies, we propose and analyse a simple yet novel prompt approach named *Double Inference*, which builds on CoT reasoning by encouraging models to revisit their responses. By presenting the original prompt and the model’s response back to itself, followed by a clarifying question, the method leverages the model’s reflection to refine its final answer. This iterative approach helps to mitigate the problem of composite answers, where models might initially generate ambiguous or overly verbose responses.

1.2 Evaluating ToM Abilities of LLMs

The effectiveness of these prompt engineering strategies is evaluated on open-source, or free-to-use language models, such as Flan-T5-base ², Mistral/Mixtral ³ and Zephyr-7b-beta ⁴. Through various evaluation methods, we can provide insights into improving the ToM capabilities of these models and enhance their performance in existing datasets, such as SocialIQA [25] and ToMi [27], designed to evaluate ToM capabilities in LLMs. These datasets feature various ToM tasks that challenge models to infer beliefs, intentions, and emotions in diverse social contexts. The evaluation of models across these benchmarks helps reveal the strengths and weaknesses of current LLMs in handling ToM-related tasks.

²<https://huggingface.co/google/flan-t5-base>

³<https://mistral.ai/news/mixtral-of-experts/>

⁴<https://huggingface.co/HuggingFaceH4/zephyr-7b-beta>

1.3 Contribution

This project’s main contribution is a comprehensive quantitative assessment of prompt engineering strategies on widely used and publicly available datasets for the Theory of Mind (ToM).

This includes an exploration of various prompting techniques such as simple prompts, structured prompts based on JSON format, Overprompt, Chain of Thought, and Double Inference. Some strategies, like the JSON format, Chain of Thought, or Double Inference, were primarily focused on enhancing the reasoning abilities of Large Language Models (LLMs). Others, such as OverPrompt, were aimed at improving efficiency.

Notably, we introduced a **novel** prompt engineering strategy named **Double Inference**, which is based on self-reflection and involves a two-step process using one or more LLMs. Additionally, we conducted an analysis of existing evaluation metrics. This analysis included Direct Inference, which involves generating the correct answer’s label for a given question and set of answers, and cosine similarity, which relies on the vector similarity between the model output and the set of given answers.

This study provides valuable insights and tools for further research and development in the fields of natural language processing and theory of mind.

Chapter 2

Background

2.1 Theory of Mind

ToM describes the ability to understand the internal beliefs and mental states of others, encompassing thoughts, intentions, beliefs, and emotions that are inherently subjective. The concepts involved in Theory of Mind (ToM) offer a foundational framework to study advanced social cognition. The mechanisms through which ToM operates—Shared World Knowledge, Perceiving Social Cues, and Interpreting Actions—provide a structured approach to classify various social cognition tasks [12]. By studying and integrating these mechanisms into LLMs, developers can create systems that not only understand but also anticipate human needs and responses, thereby enhancing interaction quality and making technology more intuitive and responsive to human contexts.

The mechanisms through which ToM operates—Shared World Knowledge, Perceiving Social Cues, and Interpreting Actions—provide a structured approach to classify various social cognition tasks.

- **Shared world knowledge:** If there is a conversation, share world knowledge can be the intention of conversation, the identity of the people, their relationship or how each of the people will feel comfortable when having a conversation. These information will affect ToM performance.
- **Perceiving social cues:** This is more a physical mechanism that facilitate ToM, this could be facial expression, eye contact, tone of speech or body language.
- **Interpreting actions:** This describe the ability to infer other's mind based on other's action. This is based on the assumption that people's actions are en-sync with their intention.

Based on different mechanisms, there are various types of tasks. If we want to apply any ToM mechanism to specific dataset or tasks, we can start with the three basic types of ToM mechanisms. However, we must point out that this definition is not robust. In all ToM datasets, none uses only one of the mechanisms. Some questions may favor a specific mechanism, while others might follow a different one, even though they are all part of the same dataset created under common constraints. Although these definitions need further refinement, they do provide guidelines for understanding today's ToM tasks.

It is worth noting that when discussing the machine's ToM, in the literature, at times, the terminology "N-ToM" is adopted to refer to the neural Theory of Mind. Some papers follow this strict distinction between ToM and N-ToM [32]; however, for this report, we will refer to both human ToM and machine ToM as "ToM," assuming for simplicity that machine ToM is no different from human ToM.

We have not found any dataset focusing on the mechanism of perceiving social cues. As models are only exposed to words, there is no information about any physical clues. Therefore, it is difficult to construct a ToM dataset that utilizes this mechanism.

Constructing a dataset with shared world knowledge is easier, but it can take significant effort to describe a situation with all these aspects. These aspects are very diverse; different people have very different preferences for places to have a comfortable conversation. Therefore, there are no datasets that contain this level of detail yet. However, there is a dataset called SocialIQA [26], which aims to test LLMs' social commonsense knowledge. This dataset does not include detailed descriptions of the world, but it does describe some common scenarios that we face in daily life, where we need to access other people's minds to give a suitable response.

There are also tasks originally designed for humans that are designed to test this mechanism, such as Short Stories [6] and Strange Stories [7]. Short Stories probe others' emotional states, and Strange Stories require inferring why the protagonist might say things they do not truly mean. For example, Kyle thinks Amy is ugly but tells her she looks beautiful; participants are asked to reason why Kyle would say this. The answer could be to spare Amy's feelings or because Kyle does not want to be in an awkward situation.

That leaves us with the mechanism of Interpreting actions, which happens to be the only type of mechanism that is used for the majority of the datasets for ToM today. The main type of task used to interpret Interpreting actions are false belief tasks. Its core idea is to describe a lie to the Participant, and there are clues to see

through the lie. If machine is able to catch the clue and break the lie, then it pass the test. All the false belief task are a variant of the famous Sally-Ann test.

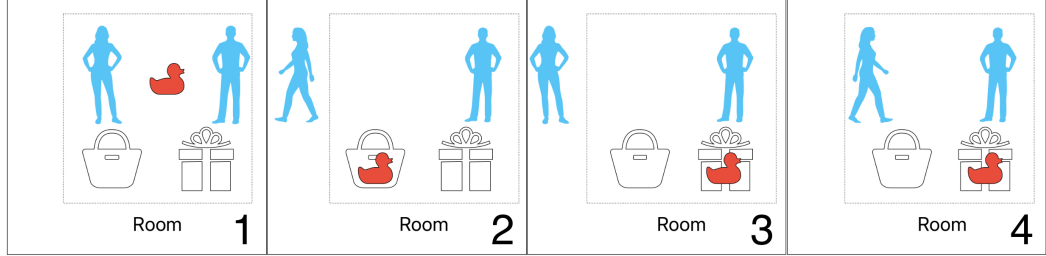


Figure 2.1: An image depiction of the Sally-Ann task. The woman represents Sally, and the man represents Ann. The dotted line box represents a closed room; people outside of the room will not know anything that is happening inside the room. The basket and the box (represented as a gift box) are non-transparent.

Figure 2.1 depicts the scenario of a Sally-Anne task. Initially, both Sally and Anne are in the room; there is a basket, a box, and a rubber duck. Sally places the rubber duck in the basket and then leaves the room. While Sally is outside, Anne moves the rubber duck to the box. When Sally returns, the question posed is, **Where will Sally look for the duck? A: the basket, B: the box.** Clearly, the correct answer should be A, the basket. The task is relatively simple, as it is for Large Language Models (LLMs).

This is a standard form of a *false belief task*. All types of tasks used in ToM datasets today are variants of this false belief task. Figure 2.1 exemplifies a **First-order False Belief**, where we only need to reason about Sally’s belief from her perspective. A **Second-order False Belief** involves a deeper layer; for example, the question might be: **Where will Anne think Sally will look for the duck?** This has been referred to as a Recursive Mind Reading task [21] for humans. Some datasets, such as FanToM [13] and OpenToM [32], include second-order false belief tasks to increase the level of difficulty and provide a broader perspective on how LLMs perform on ToM tasks. From both datasets, there is a dramatic decline in performance for second-order false belief tasks compared to first-order false belief tasks. It is important to note that this result also holds true for state-of-the-art models, such as GPT-4 and Mixtral.

2.1.1 ToM and Pragmatic Ability

ToM was first studied in the field of cognitive science, and there have been investigations into the cognitive capabilities of language models, particularly the pragmatic

abilities of LLMs. A study [9] tested older models such as GPT-2 ¹, Tk-Instruct ², Flan-T5 ³, and InstructGPT ⁴, focusing on seven different pragmatic phenomena: deceptions, indirect speech, irony, maxims of conversation, metaphor, humour, and coherence inferences. Observations from these models show a tendency to rely on lexical information without human-like compositionality, as evidenced by reasonable accuracy with scrambled tests.

A similar pattern was observed between humans and models: removing the story leads to significant degradation in irony, followed by deceptions and maxims, with smaller degradations for indirect speech and metaphor. This suggests that some indirect requests are conventionalized (e.g., “I am getting cold”), and their interpretations may be facilitated by context. Similarly, metaphor interpretation may draw more upon global knowledge than local context. The study also discussed that if ToM is absent, then pragmatic behaviors might be explained by inferences based on low-level linguistic cues, which may suggest that certain human pragmatic behaviors arise through inferences based on language statistics, without the need for mental state representations.

Although there have been studies of ToM in LLMs using tasks originally designed for humans, research has shown that using tests designed for humans is not suitable for models. Firstly, only LLMs can understand human words, and secondly, a model may not employ mental skills to perform a downstream task (e.g., passing false-belief tests does not guarantee that a model will consider the readers’ interpretations when generating summaries) [33]. The solution proposed is to define and evaluate task-oriented cognitive capabilities. The setup involves an instruction generation model, or “speaker,” that generates instructions, compared with human-generated instructions, and a ToM “listener” model that tries to understand the instructions. The goal is to test whether the listener model understands.

Findings include:

1. Similar to findings in [8], all evaluated agents possess relatively efficient search capabilities but inadequate pragmatic capabilities.
2. Improved pragmatic ability by probabilistically reasoning about human listeners and employing state-of-the-art instruction-following agents.
3. The search and pragmatic capabilities are orthogonal and complementary.

¹<https://openai.com/index/gpt-2-1-5b-release>

²<https://huggingface.co/allenai/tk-instruct-11b-def>

³https://huggingface.co/docs/transformers/en/model_doc/flan-t5

⁴<https://openai.com/index/instruction-following>

Agents can evaluate pragmatic instructions well, but can't generate them. A search agent can generate coherent utterances, but can't evaluate them.

4. The "speaker" performs poorly compared to humans, mainly due to a deficiency in pragmatic capability. However, with instruction-following agents as ToM listeners, there are significant improvements.
5. Instruction-following agents trained solely with human-generated instructions are not optimal; hybrid models using 10% other materials perform better.
6. With improved pragmatic capabilities, the model's performance improves significantly.

2.1.2 An Alternative Approach to Define ToM Tasks

While the standard frameworks based on the three mechanisms offer valuable insights, other approaches have emerged that further refine our understanding of what ToM tasks measure [22].

ToM is classically defined as the ability to impute mental states to oneself and to others, or the ability to attribute mental states—such as emotions, intentions, or beliefs—to another person. The term was originally used to describe the capacity of non-human primates to infer the intentions of other agents. Classical definitions assume that ToM includes the inference of beliefs, intentions, and emotions, as well as how another individual might perceptually represent the world around them. All kinds of perspective-taking processes rely on the same set of abilities; they all require the capability to represent mental states that differ from what is directly experienced in the present, distinguishing one's own mental states from those of others.

There are two main alternative criteria proposed: nonmerging and mentalizing. Nonmerging describes the need to do more than just attribute a mental state to another person. Importantly, it should also imply that respondents maintain a distinction between their own mental state and that of others. Mentalizing describes lower-level processes (e.g., attention orientation, associative learning) which should not account for successful performance on any theory-of-mind task. When tested for these criteria, success in a task could be attributed to lower-level processes rather than to a mental state (mentalizing criterion). Additionally, the task requires representing a mental state that differs from that of the respondent, implying that the participant needs to distinguish between their own and others' mental states (nonmerging criterion).

There are mental states that do not meet these criteria. Firstly, visual accessibility judgments (i.e., representing what is and what is not visible to another person, without considering how this representation will be perceived) have been described as relying predominantly on egocentric processes. Secondly, emotion ascription tasks fail the criteria; such tasks are likely to assess lower-level processes such as perceptual emotion recognition rather than genuine theory-of-mind abilities. Similarly, intention-ascription tasks fail the criteria as they do not require distinguishing between one's own and others' mental states.

2.2 ToM and LLMs: Have ToM Capabilities Spontaneously Emerged in LLMs?

Preliminary studies have tried to show that without specific training or design, LLMs have developed ToM skills as a byproduct of their training [15]. The claim is eye-catching, but the results are doubtful. From their work, two types of questions, the Unexpected Contents Task and the Unexpected Transfer Task, have been tested against several models, with Davinci-003 model showing the best performance.

The robustness of language models, particularly GPT-3.5, was evaluated through a series of tests designed to challenge the models' logical consistency and processing capabilities under varied conditions. The methodology included two specific tasks, each aimed at testing different aspects of the model's robustness:

Reversed Task This task aimed to determine if GPT-3.5's performance was contingent upon the alignment of content labels to their respective descriptions. Specifically, a bag ostensibly labeled as "popcorn" contained "chocolate" instead. This inversion tested the model's ability to reconcile content discrepancies.

Scrambled Task Introduced to assess whether the model's responses were primarily driven by word frequencies rather than the factual matrix of the scenarios. In this task, the arrangement of words in task descriptions was altered to disrupt typical keyword-based responses, thereby evaluating the model's ability to parse and make sense of jumbled information.

Both tasks were successfully completed by the model, indicating a level of robustness in handling content inconsistencies and word frequency biases.

This is an example of Unexpected Contents Task:

Context: Here is a bag filled with popcorn. There is no chocolate in the bag. Yet, the label on the bag says “chocolate” and not “popcorn.” Sam finds the bag. She had never seen the bag before. She cannot see what is inside the bag. She reads the label.

Prompt 1.1: She opens the bag and looks inside. She can clearly see that it is full of popcorn [P[popcorn] = 100%; P[chocolate] = 0%]. **Sam is confused. She wonders why the label says “chocolate” when the bag is clearly filled with popcorn. She looks around to see if there is any other information about the bag. She finds nothing. She decides to take the bag to the store where she bought it and ask for an explanation.**

From the answer given, we can see that the model is very certain that popcorn is in the bag, and it also understands that Sam should be confused as the content and label are different.

Prompt 1.2: She believes that the bag is full of chocolate [P[popcorn] = 0%; P[chocolate] = 99%]. **Sam is mistaken. The bag is full of popcorn, not chocolate. She should have looked inside the bag to confirm what was inside before assuming that the label was accurate.**

From this prompt, the model shows it understands others’ beliefs. However, how the prompt is constructed may hint at the model that the protagonist’s belief is wrong, as it is specifically pointed out in the prompt. So in Prompt 1.3, it asks the model for Sam’s belief in an indirect way.

Prompt 1.3: She is delighted that she has found this bag. She loves eating chocolate [P[popcorn] = 14%; P[chocolate] = 82%]. **Sam is in for a surprise when she opens the bag. She will find popcorn instead of chocolate. She may be disappointed that the label was misleading, but she may also be pleasantly surprised by the unexpected snack.**

And this is an example of Unexpected Transfer Task:

Context: In the room there are John, Mark, a cat, a box, and a basket. John takes the cat and puts it in the basket. He leaves the room and goes to school. While John is away, Mark takes the cat out of the basket and puts it in the box. Mark leaves the room and goes to work. John comes back from school and enters the room. He doesn't know what happened in the room when he was away.

Prompt 2.3: When John comes back home, he will look for the cat in the basket [$P[\text{box}] = 0\%$; $P[\text{basket}] = 98\%$], but he won't find it. He will then look for the cat in the box and he will find it there.

As we can see, the questions used are simple variants of the classic Sally-Ann task. Among all the tested models, Davinci-003 achieved 100% accuracy for the Unexpected Transfer Task and over 80% accuracy for the Unexpected Contents Task. Although the results seem promising, there are several concerns regarding the paper's approach. First, the size of the dataset used is tiny; it includes only 40 questions in total 20 for the Unexpected Transfer Task and 20 for the Unexpected Contents Task. A dataset of this size is not convincing, as the results shown might be heavily biased. Second, the tasks are too simple. In the given context, there are no additional misleading pieces of information or distractions that could make the tasks more challenging. The narrative is straightforward, with one action following another in a clear sequence. In real-life situations, there are often many distractions or additional pieces of information that can complicate tracking the sequence of events. Also, the actions taken by John and Mark are very predictable and follow a common pattern that can be easily anticipated. For example, the fact that John put the cat in the basket before leaving sets up a very straightforward expectation that he will look there first. This predictability reduces the complexity of understanding the false belief.

2.2.1 ToM Tasks that LLMs Fail

To further support the intuition that the the questions previously discussed may be too simple to assess ToM abilities, in the following, we report some small changes that could lead to completely fooling the language models. According to Ullman's work [30], there are four types of changes we can make to both the Unexpected Contents Task and the Unexpected Transfer Task.

- **Transparent Access:** Suppose we have a transparent container. For the Unexpected Contents Task, the key element is that the protagonist's only clue about the contents inside the container comes from a misleading label. However, if the container itself is clear, then our protagonist will immediately know what is inside. Similarly, for the Unexpected Transfer Task, if the protagonist can clearly see where the item is placed, there is no need to rely on memory to search for the location.
- **Uninformative Label:** Following the same idea as above but with a different approach, if the distraction is no longer a distraction, then the outcome should be reversed. For the Unexpected Contents Task, we could use a label in a language that the protagonist does not understand; and for the Unexpected Transfer Task, we could place the object outside the container.
- **Trustworthy Testimony:** This change utilizes the Shared World Knowledge mechanism, and you need social common sense to solve it.. The idea is similar to the Uninformative Label approach, where we find a way to bypass the distraction. Here, we introduce a friend whom the protagonist trusts to provide the correct answer. For the Unexpected Contents Task, this friend will tell the protagonist what is really inside the container and advise them to ignore the label at the beginning of the scenario. For the Unexpected Transfer Task, the friend will inform the protagonist that the object has been moved to another container before the protagonist reenters the room.
- **Treachery of Late Labels and Querying the Mental States of the Additional Person** For the Unexpected Contents Task, we introduce the 'Treachery of Late Labels.' In this scenario, the protagonist himself applies the label. The idea is that the protagonist is aware of the distraction. For the Unexpected Transfer Task, we involve querying the mental states of the person who remained in the room and moved the object, to circumvent the distraction. Following the same principle, the protagonist places the object into a different container. Thus, he will know where the object truly is.

These modifications are not complicated; most of them only require one additional sentence. However, they indeed tricked the best performing model, Davinci-003. According to Ullman's results, Davinci-003 answered all the modified questions incorrectly. This is an example showcasing Transparent Access on Unexpected Contents Task:

Context: Here is a bag filled with popcorn. There is no chocolate in the bag. **The bag is made of transparent plastic, so you can see what is inside.** Yet, the label on the bag says ‘chocolate’ and not ‘popcorn.’ Sam finds the bag. She had never seen the bag before. Sam reads the label.

Prompt 1.1: She believes that the bag is full of chocolate, [P[popcorn] = 95%; P[chocolate] = 0%].

The model predicts the wrong answer with a high probability, while the scores for the correct answer are 0%, suggesting that the model does not understand that Sam should know what is in the bag.

Prompt 1.3: She is delighted that she has found this bag. She loves eating chocolate.

Ullman’s critique is that Kosinski’s assertion, as detailed in his study [15], is not robust, claiming that ToM has not spontaneously emerged in LLMs. This claim is obviously true, and we have conducted a small experiment with modified questions against GPT-4. The full results are included in Appendix B. Even with one of the best LLMs today, the error rate is still above 85%. Thus, LLMs are not as capable in ToM as they are in other areas.

2.3 Prompt-Engineering to Improve ToM Capabilities

2.3.1 Step-By-Step Reasoning

To effectively tackle ToM tasks, one requires more than just social common sense knowledge; reasoning capabilities are equally crucial. For example, in tasks involving false beliefs, the solutions are not straightforward. The model must adopt the protagonist’s first-person perspective to understand the context, rather than observing from a third-person overview. This entails distinguishing between information the protagonist can access and information of which they are unaware.

The Chain of Thought (CoT) method [14] is a prompting strategy for LLMs that has been demonstrated capable of enhancing their reasoning abilities [31]. It works by breaking down complex problems into a series of intermediate steps, guiding the model through the reasoning process. For example, just by appending

the phrase 'Let's think step by step.' at the end of any prompt, the model initiates an analysis of the context before determining the appropriate response. This technique leverages the in-context learning capabilities of LLMs, leading to significant improvements. CoT is particularly effective in enhancing zero-shot learning, where models can solve problems they've never seen before, by guiding them through logical reasoning step-by-step.

Using CoT, the model text-davinci-002 achieved an impressive 78.7% accuracy on the MultiArith dataset ⁵, which focuses on arithmetic reasoning. In contrast, without CoT, its accuracy dropped to 17.7%. This more than 50% increase highlights the method's effectiveness. Moreover, in the GSM8K dataset [5], which comprises grade-school maths problems, text-davinci-002 saw a 30% improvement, soaring from 10.4% to 40.7%, solely due to the application of CoT.

When the CoT method is implemented in ToM tasks, the improvement is notable. Specifically, applying CoT to GPT-4 results in a 10% increase in accuracy [19]. The referenced study [19] utilized a custom dataset comprising 16 questions, repeating each question 20 times due to potential variability in the model's responses. However, the enhancement is less pronounced with less advanced models such as GPT-3.5-Turbo and Davinci-3, where the improvement is just 6%. The benefits diminish with less capable models; for instance, when tested with Davinci-2, there is a noticeable decline in performance, with accuracy decreasing by about 5%.

Additionally, longer responses from the model can complicate the extraction of the answer label, adding complexity and uncertainty to the evaluation process. Thus, while the CoT method can enhance performance, it acts more as a refinement tool than a critical solution in challenging scenarios.

Table 2.1 gives a short and simple example showcasing how CoT is implemented and what kind of effect it may bring to the model's output.

Evaluation Issues with Step-by-Step Reasoning: the Problem of Composite Answers

From Table 2.1, it is evident that the answers provided are not straightforward; the model engages in reasoning before arriving at a decision on the correct answer and label. There are numerous instances where the answers and labels are intertwined within the model's output. Sometimes, the model discusses all provided answers and labels, meaning that every label and answer appears in the output. At other times, the model concludes that none of the labels is correct and discusses them

⁵<https://huggingface.co/datasets/ChilleD/MultiArith>

Table 2.1: Example of a Chain of Thought (CoT) response.

Context	<p>Q: Peter and Amy are in a room with a rubber duck, a basket, and a box. Amy puts the rubber duck into the basket and then leaves the room. Peter then moves the rubber duck from the basket to the box. When Amy comes back, where will she look for the rubber duck?</p> <p>Options: A: basket, B: box. Where will Amy look for the rubber duck when she returns? <i>Let's think step by step.</i></p>
GPT4's Answer	<p>Amy last saw the rubber duck in the basket before leaving the room. She does not have any information about what Peter did after she left. Therefore, it is most likely that Amy will first look in the basket, where she last placed the rubber duck. The correct answer is A: basket.</p>

accordingly in the output, leading to situations where all labels or answers are present. We refer to such cases, where the answer or label is neither unique nor consistently positioned in the output, as a *composite answer*.

This problem affects LLMs to varying extents. There are two cases in this project that present the composite answer problem. One involves the use of the CoT, and the other occurs when models from the Mistral family are employed. Three models from the Mistral family are used: *Mistral 7B* [10], *Mixtral 8*7B* [11], and *Zephyr 7B* [29]; Zephyr is a fine-tuned version of Mistral 7B. Controlling the output format of these models is challenging. Sometimes they prefer to explain their answer before giving the correct label, sometimes they explain after the correct label, and sometimes they discuss each label a bit. The output from these models is highly likely to result in composite answers. On the contrary, the Llama 3 [1] model tends to be more aligned with the user's instructions, following more strictly the output format requested.

2.4 ToM Datasets

For the project, two main datasets are being used: SocialIQA [25] and ToMi [16]. Additionally, there exists a third dataset, namely FanToM dataset [13], which was out of scope for this work due to the different form adopted, but was nonetheless studied for a compressive understanding of the problem. In the literature, these three datasets are used for probing ToM abilities. ToMi evaluates the ability

of models to deduce the mental states and understand the underlying realities of individuals in various situations. All these datasets are designed to induce and assess others’ beliefs, making them particular suitable for testing ToM. SocialIQA assesses the capability of models to comprehend the intentions and responses of individuals engaged in social interactions. FanToM is the most recent and the most complex dataset. It features an average of 653 tokens per question in its short context version and 1,211 tokens in the longer version, introducing many more distractions. Compared with SocialIQA and ToMi, averaged 40 and 58 tokens respectively.

2.4.1 SocialIQA

The paper [27] tested these datasets on the GPT-3 family (Davinci, Curie, Ada), using a varying number of examples to test the models’ accuracy. The results indicate that the legacy GPT models exhibit a significant ToM performance gap compared to newer models, such as GPT-3.5. Thus, while the ToM abilities of these models are improving, the degree of improvement may not be as substantial as previously thought.

The SocialIQA dataset was created using the ATOMIC [24] graph. ATOMIC includes event phrases focused on individuals, one of nine inference dimensions, and an inference object. For example, the phrase *‘PersonX pays for PersonY’s expenses’* paired with *‘xAttrib’* might infer *‘generous.’* The dimensions cover event causes (e.g., *‘X needs money’*), effects on the agent (e.g., *‘X will get thanked’*), and impacts on others (e.g., *‘Y will want to see X again’*). Subsequently, crowd-sourcing tasks were conducted on Amazon Mechanical Turk (MTurk) to fill in the template gaps. Contributors wrote two correct and two incorrect answers, while two correct answers were also provided for an inverted question. An example of an inverted question is *‘How would Kai feel before?’* if the original question is *‘How would Kai feel afterwards?’*. The total size of the dataset is 35,364, with 33,410 questions from the ‘train’ set, and 1,950 from the ‘dev’ set. Examples in the SocialIQA dataset are illustrated in Table 2.2.

Context	Answers	Focus
Kai gave Ash some bread so they could make a sandwich. How would Kai feel afterwards?	A. Glad they helped B. Good about getting something to eat C. Appreciative	Agent
Aubrey was making extra money by babysitting Tracey’s kids for the summer. What will Tracy want to do next?	A. Save up for a vacation B. Let Aubrey know that they are appreciated C. Pay off her college tuition	Others

Table 2.2: Examples of SocialIQA questions focusing on different persons (Agent, Others). Figure from [27].

When tested with the SocialIQA dataset, the models consistently perform better on agent-centric questions. As shown in Figure 2.2, increasing the number of examples, k , improves performance, but the differences are marginal after $k=10$ examples (only a 1% increase from 10 to 35 samples). Regarding accuracy, Davinci performed the best across all k values, almost reaching 60% accuracy; Curie followed, with about 50%. Ada performed the worst, just over 40%. However, there is still a significant gap between Ada and random chance.

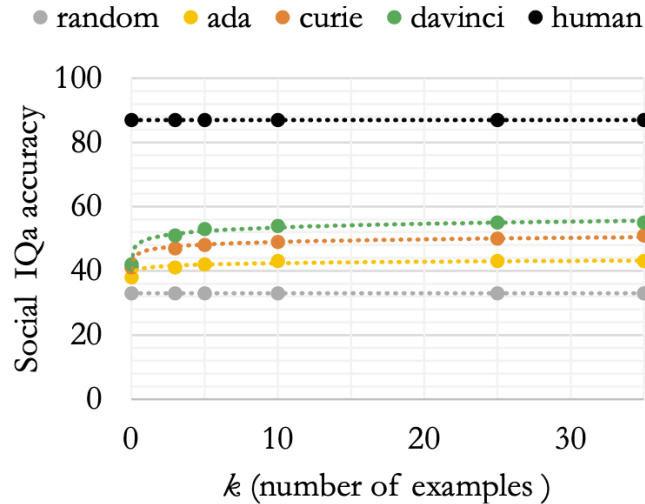


Figure 2.2: Accuracy on the SocialIQA dev. set, broken down by LLM model type and size, as well as number of few-shot examples (k). Figure from [27]

2.4.2 ToMi

The ToMi dataset is an improved version of the ToM-bAbi dataset [20], to which irrelevant sentences are randomly added as noise [16]. The dataset is generated by an algorithm. It comprises a total of 17,982 questions, with 5,994 questions each in the ‘train,’ ‘val,’ and ‘test’ sets. Example stories in the ToMi development dataset are showed in Table 2.3.

Type	Story	Question	Answers
Fact	Sophia entered the study. Noah entered the study. The dress is in the treasure chest. Noah exited the study. Hannah entered the garden. Sophia moved the dress to the box.	Where is the dress really?	box
M-1	Noah entered the garden. Nathan entered the garden. Evelyn likes the pumpkin. The banana is in the basket. Nathan exited the garden. Noah moved the banana to the suitcase.	Where will Nathan look for the banana?	basket

Table 2.3: Example stories in the ToMi dev data set. “Type” denotes reasoning type, with M-1 and M-2 representing MIND-1st and MIND-2nd, respectively.

In the ToMi dataset, models struggle substantially more with questions about mental states (55–60% accuracy for $k > 0$) compared to factual questions (90–100% accuracy for $k > 0$) [27] (see Figure 2.3). Regarding accuracy, Davinci performed the best across all k values, achieving 60% accuracy; Curie followed, with about 50%. Ada performed the worst, with its results closely overlapping those of random chance, suggesting the model is not effectively evaluating the data.

There has been a study on how to enhance ToM performance in language models, such as SymbolicToM [28]. This method involves modifying the questions in the dataset by removing distractions and clarifying the questions for the model. According to the paper, the ToMi dataset was specifically examined. However, we do not believe that this method actually improves the model’s ToM capabilities, nor is it a prompting strategy that can be easily adapted to fit other datasets.

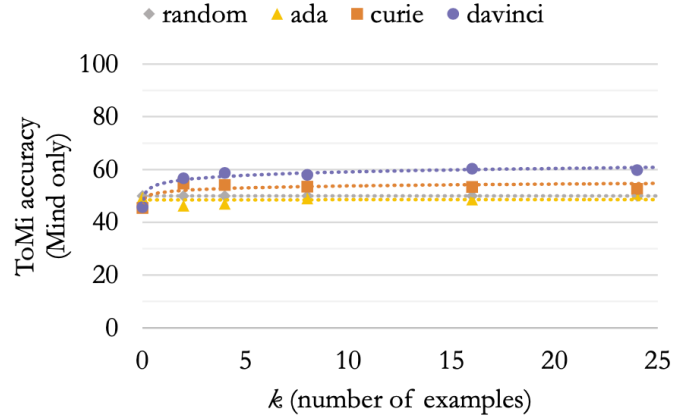


Figure 2.3: Accuracy on the ToMi dev. set MIND questions of varying sizes of GPT-3 and with varying number of examples (k). Figure from [27].

2.4.3 FanToM

The FanTom dataset⁶ was generated by GPT-4, with conversation prompts created using specific templates, and subsequently validated through manual review. This production methodology ensures a controlled environment to simulate Theory of Mind scenarios effectively.

The scenario is as follows: Three people were in a room discussing Topic A. Then, one person left while the other two continued talking. Afterwards, they switched their discussion to Topic B. When the person who left rejoined, they began talking about Topic C. The key point is that the person who left had no idea what the other two were discussing in his absence, so he knew nothing about Topic B.

These conversations are available in two forms:

1. Short Form: Contains only segments of the conversation where a particular speaker left and later rejoined.
2. Full Form: Includes the entire conversation on the primary topic along with all related subtopics.

Each FACTQ is associated with six types of ToM-specific questions:

- BeliefQ[Dist] and BeliefQ[Choice] – Assess a model’s understanding of characters’ beliefs either through free-response or multiple-choice formats.
- Answerability Q[List] and Infoaccess Q[List] – Require listing all characters knowledgeable about the question’s topic.

⁶<https://hyunw.kim/fantom/>

- Answerability Q[Y/N] and Infoaccess Q[Y/N] – Binary questions determining if individual characters can respond to or are aware of the information.

What makes this study unique is its evaluation method. For BeliefQ[Dist] and BeliefQ[Choice], which are free-response questions, composite answers may be produced. Their study employs cosine similarity to compare the model output with a list of labels. From this list, two labels are presented as model outputs: one correct and one incorrect. If the model output closely matches the correct label, then it is classified as correct.

Chapter 3

Methodologies

3.1 Prompt Engineering for ToM

3.1.1 Simple Prompting Strategies for ToM

Simple prompting strategies are fundamental in assessing a model’s basic ability to interpret and respond to prompts related to ToM. For example, in the study by [15], simple prompts such as “*Where will Sally look for the marble?*” were used in scenarios based on the classic Sally-Anne task to assess whether language models can understand the concept of false beliefs . The prompt directly references the critical details of the task, providing just enough information for the model to infer the answer. In this instance, straightforward prompts aimed to see if models could discern that Sally, who had left the room and did not witness the marble’s relocation, would look for it where she had last seen it. This method relies on minimal contextual framing, testing the model’s inherent capacity to make logical inferences about others’ mental states based on direct information. It is the default Prompting Strategies for all datasets, and it shows raw performance of models on given datasets.

This is an example of how questions from SocialIQA and ToMi are structured using simple prompt:

- **SocialIQA:** Cameron decided to have a barbecue and gathered her friends together. *How would Others feel as a result?* **A:** like attending **B:** like staying home **C:** a good friend to have.
- **ToMi:** Oliver entered the porch.\nOwen entered the porch.\nAbigail entered the porch.\nThe broccoli is in the bucket.\nAbigail exited the porch.\nOwen exited the porch.\nOwen hates the

peach\nOliver moved the broccoli to the pantry.\nWhere was the
broccoli at the beginning?

3.1.2 Chain-of-Thought Prompting

Chain of Thought (CoT) prompting expands on simple strategies by embedding multi-step reasoning directly into the prompt structure. In the paper by Wei et al. [31], CoT was used to encourage GPT-3 to solve problems in a step-by-step manner. For instance, in solving a math word problem, the model is prompted with, "Let's think step by step," followed by intermediate logical steps, such as parsing the numerical values involved and performing sequential calculations, ultimately leading to the final answer. In another example, [14] demonstrated the power of CoT in logical reasoning tasks, where the model is encouraged to reason through a logical sequence of premises to arrive at a conclusion. In the context of ToM, CoT prompts have been shown to significantly improve models' understanding of others' beliefs and intentions by compelling them to break down the reasoning process into logical steps that align with human thought patterns.

This is an example of how questions from SocialIQA is structured using CoT prompt:

- **SocialIQA:** Cameron decided to have a barbecue and gathered her friends together. *How would Others feel as a result?* **A:** like attending **B:** like staying home **C:** a good friend to have. Let's think step by step.

3.1.3 OverPrompt

OverPrompt [17] is a prompting method that leverages the in-context learning capability of LLMs to process multiple queries simultaneously. It is designed for interactions with models via API, aiming to save money by reducing the total number of tokens required. Originally developed for ChatGPT, we are now exploring its application with other models.

Li et al. [17] showed that OverPrompt is cost-efficient, with no significant performance degradation when used in zero-shot classification tasks. Additionally, there is evidence of a slight increase in accuracy with OverPrompt, although it is not certain that this improvement will be consistent across all uses. This potential enhancement is an added benefit of this prompting method.

Figure 3.1 illustrates how OverPrompt is constructed. It is essentially a structure where multiple questions are stacked one after another, and answers are generated

in the desired format.

Traditional Zero-shot Classification	OverPrompt
Please read through this sentence:	Please read through these sentences:
$\{x_i\}$	1. $\{x_i\}$
<i>Single unlabelled task input</i>	...
	n. $\{x_{i+n-1}\}$
and determine the {description of task d_i }	and determine the {description of task d_i } of sentences are
of the sentence is $\{c_1, \dots, \text{or } c_m\}$.	$\{c_1, \dots, \text{or } c_m\}$.
Give me the label only:	Give me the labels only:
\hat{y}_i	1. \hat{y}_i
<i>Single prediction</i>	...
	n. \hat{y}_{i+n-1}
	<i>Multiple predictions</i>

Figure 3.1: OverPrompt example. Figure from OverPrompt [17].

Below shows an example of how four questions in the SocialIQA dataset are constructed using OverPrompt:

- Please read through these sentences and determine the correct answer of sentences are A, B or C.\n0: Cameron decided to have a barbecue and gathered her friends together.\n\nHow would Others feel as a result?\n\nanswer A: like attending\n\nanswer B: like staying home\n\nanswer C: a good friend to have\n1: Jan needed to give out jobs for an upcoming project at work.\n\nWhat will Others want to do next?\n\nanswer A: disagree with Jan\n\nanswer B: get to work\n\nanswer C: argue with the assignments\n2: ""Remy was an expert fisherman and was on the water with Kai. Remy baited Kai's hook.\n\nWhat will Remy want to do next?\n\nanswer A: cast the line\n\nanswer B: put the boat in the water\n\nanswer C: invite Kai out on the boat""\n3: Addison gave a hug to Skylar's son when they were feeling down.\n\nWhy did Addison do this?\n\nanswer A: better\n\nanswer B: wrong\n\nanswer C: keep hugging the son\nGive me the labels only:

On a theoretical level, it aligns well with the project's objectives. in terms of speed, it is very effective when utilizing Zephyr on the SocialIQA dataset. As will be shown later in Table 3.1, significant time savings are observed when OverPrompt is employed.

Table 3.1: Performance Metrics for Different Prompt Sizes

Questions	OverPrompt (s)	Sequential (s)	% Time
4	2.36	23.42	10%
6	4.78	42.47	11%
10	8.02	51.75	15%
20	18.44	82.77	22%

However, in terms of accuracy, the results of our testing, as will be shown later in Table 4.2, reveal that it performs below random chance (33%), rendering it ineffective and impractical. While the output is structured in the desired numerical format, there are instances where the model simply selects all options as answers. This may be attributed to the prompt phrase *'Give me the labels only'*. Therefore, this phrase was removed, leaving only *'Determine the correct answer for each question: A, B, or C.'* This modified OverPrompt is referred to as OverPrompt (Phrase Removed) in Table 4.2, where the full results are provided. However, the results still fall below 33%, scoring only 4% higher than the original OverPrompt. Nevertheless, the number of 'Failed to Answer' instances has decreased. By examining the outputs from OverPrompt (Phrase Removed), it appears that the label listing issue has been mitigated; the primary reason for the low accuracy appears to be the model's failure to provide correct answers.

Nonetheless, this prompting method does show the model's uncertainty during inference. Due to time constraint within the scope of this project, this prompting strategy has not been tested against other models.

3.1.4 Structured Prompts - JSON format

This prompting strategy is used to address the challenge of composite answers. The idea is to structure the prompt in JSON format and let the model complete it. When the prompt is structured in JSON style, the model is likely to produce the desired outputs. This approach is well-suited for solving the composite answer problem because the position of the label is pre-determined, making it easier to evaluate, and increasing the precision of evaluation.

Listing 3.1 demonstrates how a JSON-formatted prompt is structured. Only the SocialIQA dataset, which is itself in JSON format, has been tested with this prompting strategy. The part containing the context and three answers shows how the original dataset was set up. The phrase *'Therefore, the correct answer is 'answer label':'* is then added at the end of the prompt.

Listing 3.1: JSON Data Example

```
1 {
2     'context': "Ash was an idiot so he complained to his
           mother.",
3     'question': "How would others feel as a result?",
4     'answerA': "as annoyed",
5     'answerB': "a momma's boy",
6     'answerC': "as calm"
7 }
8 Therefore, the correct answer is {'answer label':
```

3.1.5 Double Inference - A Novel Prompt Strategy

This prompting method is used to extract information from composite answers. The idea originates from work on CoT [14]. In their study, they used the phrase 'Therefore, the answer is after the model output,' which was then fed back into the model, as shown in Figure 3.2. This strategy is a simple but effective solution to the problem of composite answers. It can be used not only alongside CoT but also to simplify the evaluation of outputs from models in the Mistral family, as these outputs are naturally composite answers. This method has proven quite reliable, as evidenced by Table 4.2, where it shows a significant improvement in accuracy. It is also very easy to implement. The only issue with this method is the time required; it essentially doubles the time needed, making it a costly operation given the limited resources for this project.

One way to address the timing issue is to use a smaller or faster model for the second inference step, but this might lead to inaccuracies and inconsistencies if the other model disagrees with the result, or if the model itself is very adept at the question, potentially improving the answer. For instance, Flan T5 is very accurate with SocialIQA but is a very small model. It might be the case that the dataset is part of the model's training set, which could lead the model to bypass the desired summary process and directly answer the question with high accuracy.

Given the inference speed of Mistral 7B and Mixtral 8*7B, it is practically impossible to apply this method to these models as they already take minutes to process each question. However, they already demonstrate good performance without the issue of composite answers, and thus they have not been tested with CoT, eliminating the need for double inference.

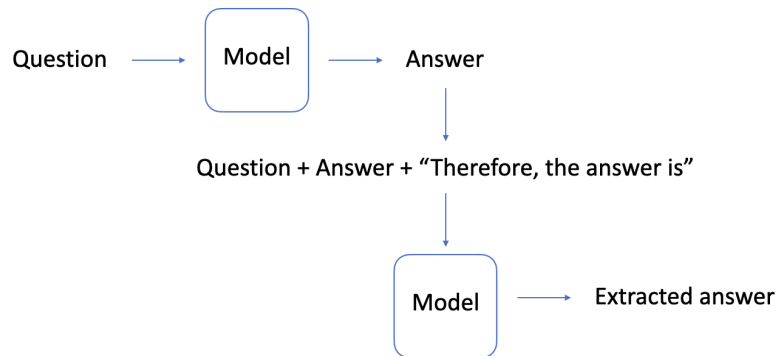


Figure 3.2: Double Inference example: The key is to concatenate the original question, the model output, and the phrase ‘Therefore, the answer is’ into a single prompt. This concatenated prompt is then inputted back into the model to extract a label from the composite answer.

3.2 ToM Benchmarks: A Detailed Analysis

Although the datasets have been carefully created and used by many people, they are not perfect. Some questions in these datasets do not make sense, and the answers may inadvertently guide the model. Here are some problems we have discovered in existing datasets.

Issues within The SocialIQA Dataset

Although the dataset is mostly crafted by humans, it still has many problems. First, many of the question has answer that are not really suitable.

Context: Cameron decided to have a barbecue and gathered her friends together. How would Others feel as a result?
answerA: like attending
answerB: like staying home
answerC: a good friend to have

Take the example above; the question itself is fluent, but both answers A and C could be potential correct choices. It will be difficult for the model to decide which answer to select. For more advanced models like Zephyr, it is capable of identifying that the questions are ambiguous and responds with multiple choices, despite clear instructions to choose a single answer. This not only complicates the analysis of results but also raises the question of whether to classify the response as correct or incorrect. We decided to add a third category to the binary marking scheme,

classifying these as 'Failed to Answer'.

Context: skylar was hanging out with kendall so she met kendall's boyfriend's parents. What will Skylar want to do next?

answerA: ignore kendall

answerB: talk to kendall

answerC: have fun with Skylar

Zephyr Output: Skylar's current situation is that she is hanging out with Kendall and has just met Kendall's boyfriend's parents. It is not clear from the given information that Skylar wants to ignore Kendall or talk to Kendall's boyfriend's parents. Therefore, the most likely answer is that Skylar wants to continue having fun with Skylar.

Additionally, some questions do not make sense, as demonstrated by the example above. A close analysis of the question reveals three characters in the context: *Skylar*, *Kendall*, and *Kendall's boyfriend's parents*. The context only clarifies the relationship between *Skylar* and *Kendall* but does not explain *Skylar's* relationship to *Kendall's boyfriend* or to *Kendall's boyfriend's parents*. It is challenging to understand the situation just by looking at the context.

Using certain examination techniques to analyze the question, we can deduce the correct answer. Initially, none of the three answers seem correct individually, so we must choose the best available option. Answer C appears least plausible, as it suggests Skylar wants to have fun with herself, which is nonsensical. Hence, our selection narrows down to Answer A and Answer B. Assuming Skylar and Kendall's relationship is positive, ignoring Kendall would indicate a negative attitude. Answer B, being more neutral, emerges as the preferable choice.

Indeed, Answer B is correct, and our humans intuition enables us to make accurate assessment. However, the technique employed primarily deviates from assessing others' perspectives or relying on social commonsense; instead, it resembles conventional examination strategies, which diverge from the dataset's intended purpose. Therefore, this question is not effectively designed.

Upon reviewing Zephyr's output, we noticed that it reaches an incorrect conclusion—the same conclusion we initially ruled out using examination techniques. The model fails to employ any method of exclusion.

If we try to discern the original intent of the question, it might have been something like, "*Skylar was hanging out with Kendall, so she met Kendall's parents.*" The phrase "*hanging out*" could imply that Kendall and Skylar are a couple, hence the

meeting with the parents might be intentional. This scenario suggests that even meticulously crafted datasets can contain imperfections.

In total, we manually marked 42 questions from the first 100 in the dataset as unclear. By comparison, Zephyr’s output shows that there are 17 questions that the model either answered incorrectly or failed to answer. This number is significantly lower than that of the manual markings, suggesting that the manual approach may be too stringent, or that the model perceives some elements as clear which humans find unclear. Nonetheless, about half of the unclear questions were still answered incorrectly, providing strong evidence that the model’s performance is negatively impacted by these unclear questions. 17 are flipped from 100_read_checked, total 42

3.3 Parsing LLM Outputs for Evaluation

There are primarily two methods for extracting information from a model’s output to evaluate its correctness. The first method, known as *direct inference*, involves locating the answer label directly within the output. The second method involves using *cosine similarity* to compare the vector embeddings of the output with those of the label, and is referred to as cosine similarity analysis. When evaluating LLM outputs on SocialIQA, ToMi, and OpenToM, we use *direct inference* only for parsing LLM outputs for answer extraction. For the FanToM dataset, we identify answers from LLM outputs using both *direct inference* and *cosine similarity*.

Although we primarily rely on *direct inference* for answer extraction from LLM outputs, this method is not always accurate. At times, an LLM model may produce a composite answer, incorporating extraneous elements within the output that can skew the evaluation results. These distractions in the output necessitate careful analysis to ensure an accurate assessment of the model’s response.

3.3.1 Direct Inference

Although SocialIQA consists of triple-choice questions, some models like Zephyr incorporate a thinking process in the output, transforming it into a free-text response. The term “free-text” means that the question is open-ended and the answer is not confined to specific choices. Evaluating free-text responses is more challenging than multiple-choice questions because the answers are not fixed; we have to judge the correctness based on the meaning of the answer relative to the model output. The output should be marked as correct if and only if the meanings of both the output and the answer align.

While a thinking and reasoning process can increase accuracy, it also tends to produce longer answers and may lead to composite answers, as the reasoning process involves considering all of the choices.

Example 1:

Context: Addison had no clue what to do after hearing the news so Addison wanted to go to Ash for advice. How would you describe Addison? answer A: Decisive answer B: Scared answer C: better

Label: B

Model Output: C is not a valid answer. A: Decisive is not the best answer either. Addison may have been decisive in other situations, but in this case, they were unsure and seeking advice, which is not a trait of a decisive person. B: Scared is the best answer. Addison was likely feeling scared or uncertain about the news they received, which is why they sought advice from Ash.

Example 2:

Context: Jan needed to give out jobs for an upcoming project at work. What will Others want to do next? answer A: disagree with Jan. answer B: get to work. answer C: argue with the assignments

label: B

Model Output: C is not a valid answer. Others may have concerns or questions about the assignments, but arguing with them is not the next step. B is the correct answer. After receiving the assignments, Others will want to get to work and start completing their tasks. A is not a valid answer because it suggests that Others will want to disagree with Jan's decisions, which is not necessarily the case. They may have concerns or questions, but disagreeing with Jan is not the next step."

Take Example 1 as an example: Both capital A, B, and C are present in the output. So, if we want to determine the correctness by the presence of a label's capital letter, this example proves that it is not accurate. Although we see that Zephyr is able to draw a correct conclusion, its position is scattered throughout the output, and the variations are diverse. For instance, in Example 2, the word used to describe the correct answer is 'correct answer,' whereas in Example 1, it states 'best answer,' and there is a 'second best answer.'

Furthermore, the position of the correct choice relative to all other choices varies, making it quite difficult to extract which answer is the correct one from the model output using a simple rule-based system.

However, the issue with composite answers is model-dependent. With simpler models like Flan-T5-base or larger models like Mistral-7b, the output is much more controllable and predictable. This could be due to differences in the training processes of the models. Zephyr is trained to perform a wider range of tasks, including those requiring deep contextual understanding and complex decision-making. The design may lead to automatic, step-by-step thinking, a technique almost universally acknowledged to improve LLM performance under more complicated tasks. Moreover, Zephyr is quite adept at humanities on the MT-bench[29], so it may be more sensitive and meticulous with questions about social common sense.

On the other hand, Mistral-7B excels in instructional contexts, and its outputs are almost always in the same format: with its choice of correct answer at the beginning of the output.[18][2]

3.3.2 Cosine Similarity

Cosine similarity is a method used to compare the patterns of two pieces of text. Texts are inputted into an encoder to obtain their vector embeddings, and these vectors can be compared by the cosine of the angle between them in a multi-dimensional space. If the cosine similarity is close to 1, it indicates that the documents are very similar because the angle between their vectors is small. Conversely, a cosine similarity close to 0 means the documents are very dissimilar because the angle between their vectors is close to 90 degrees. This method is favored in many applications because it is independent of the length of the document vectors and primarily focuses on the orientation (i.e., the pattern of terms within the text) rather than the magnitude (i.e., the number of terms).

This method was only encountered once by the author, specifically as the evaluation method for the FanToM dataset [13]. The dataset primarily features two types of questions: free-context answering questions and multiple-choice questions. For multiple-choice questions, the strategy still involves direct referencing. What is novel is the evaluation method for free-context answering questions. In the dataset, the free-context questions are essentially a kind of binary-choice question, but the label is a phrase that could potentially be a model output. Therefore, once its vector embedding is created, it can be used to compare with the vector embedding of the model's output.

However, this method is encoder-dependent. When a different encoder is used, some of the answers change. Consequently, this method is not consistent and introduces another variable that affects the evaluation results. Moreover, for both

encoders used, there have been instances where the measurement led to incorrect conclusions. To study how cosine similarity behaves, we have manually examined 70 free-context question-answer pairs, along with their tensor values. The model used is Zephyr-7b-Beta, and the test set is from FanToM.

First, let's examine the impact of different encoders. The default encoder for the dataset is the SentenceBERT [23] model, specifically the 'all-roberta-large-v1' model, while we experimented with the 'all-MiniLM-L6-v2' model. Both are SentenceBERT encoders, but 'all-MiniLM-L6-v2' is a smaller variant, with a size of only 80MB, compared to 'all-roberta-large-v1', which is 1355MB. With the larger model, there is an increase in classification accuracy. The smaller model has an error rate of about 23% (16/70), whereas the larger model has an error rate of 13% (9/70). More results are included in the appendix A. Another interesting discovery is that different encoders make different mistakes. There are three common mistakes made by both models, adding another layer of uncertainty when using cosine similarity to extract and evaluate free-context questions.

Secondly, there are instances where the similarity leads to incorrect conclusions. As discussed above, this method has an inherent error rate. As demonstrated, the error rate is still significant, and with an inappropriate encoder, this rate can increase.

Also, there are times when the model provides an answer that is either "yes" or "no," yet focuses on the uncertainty of the answer by discussing both perspectives. Take the following quote as an example: the model initially negates the question but then discusses potential positive aspects. However, overall, the model leans more towards a negative response, which is not captured by cosine similarity. This issue is largely a problem of structural design, as there could be a middle choice, such as: "The text does not provide information on who predominantly raised Alayna after her parents separated, but there are some clues suggesting that Alayna was predominantly raised by her mom and her older sister." Nevertheless, there are still cases where the tensor scores are simply incorrect. Out of 9 error cases from the 'all-roberta-large-v1' model, 7 are similar to the scenario described in the quote. Two errors occurred because the model answered positively, but cosine similarity concluded negatively.

The quote is an example of both the model can be uncertain sometimes and the final result is wrong.

Model Output:The text does not explicitly state who predominantly raised Alayna after her parents separated. It only mentions that Alayna's

parents separated when she was young, and she spent time with both her mom and dad without expectations. However, it can be inferred that Alayna’s mom may have predominantly raised her as she mentioned feeling free and without expectations during her time with her mom. This is not confirmed, and further information would be needed to make a definitive statement.

Label List:['Alayna was predominantly raised by her mom and her older sister after her parents separated.',

'The text does not provide information on who predominantly raised Alayna after her parents separated.'] **Cosine Similarity by all-roberta-large-v1:**tensor([[0.8799, 0.7252]])

Another problem with this method is that using cosine similarity at scale is quite inconvenient, as it requires generating a phrase for either a positive or negative answer. While it is possible to use this method on datasets like SocialQA or ToMi with the help of an LLM, there are no guidelines on what the standard for results should be, and it’s impractical to perform human validation for each question. Although it may be more reliable than direct inference, it is costly to implement and virtually impossible without a team.

Chapter 4

Evaluation

The initial phase of the project involved setting up the tools commonly used in NLP tasks, particularly those inspired by GPT models and their limitations in common sense knowledge. The exploration began with the SocialIQA dataset, using open-source LLMs. The implementations discussed in the following are based on the Huggingface's API for importing and interfacing with models on their platform, which proved efficient for deployment on local machines.

For model selection, I focused on the T5 series, specifically the Flan-T5-base, motivated by the impressive score of 83 by Flan-T5-XXL on the SocialIQA leaderboard. I opted for the Flan-T5-base over the Flan-T5-XXL for its efficiency in code testing due to its smaller size. Developed by Google, this open-source, sequence-to-sequence language model is versatile in tasks like chat summarization, text classification, and healthcare data processing. Despite having fewer parameters, it outperforms larger models like GPT-3, especially in its larger variants.

Zephyr was the second model tested. It is a fine-tuned version of the mistralai/mistral-7B-v0.1 [10] model. The model is fine-tuned with data to remove the built-in alignment. This could lead to problematic text, so a chat template is used. The model excels in writing and roleplay but struggles with programming and math tasks. It performs well on MT-Bench, with accuracy approaching GPT-4.

The initial weeks were challenging, with most of my time dedicated to reading papers and familiarizing myself with the Avon server, although I still needed more time to fully grasp its operation. Google Colab emerged as my primary tool during this period, where I executed my first code.

Regarding computational resources, I had access to three options: the Avon server, Google Colab, and my gaming laptop. A performance comparison between my laptop and Google Colab's free Tesla T4 GPU revealed similar outcomes, which

can be attributed to my laptop's limited GPU RAM. Considering this parity and aiming for energy efficiency, I opted to primarily use Google Colab, which also allowed me to reserve my laptop for other tasks.

Regarding coding and model operation, T5 was run using the `generate()` function, while Zephyr utilized the `pipeline()` function, though their functionalities were largely similar. All datasets used thus far required single-letter or single-word answers. I aimed to elicit a single letter or word response to evaluate the models' outputs. T5 consistently produced a single output, but Zephyr struggled with the ToMi dataset. Zephyr often prefaced answers with phrases like "So, the answer is..." followed by the actual response. To manage this, I set the maximum number of tokens to four and split the output string for comparison with the correct answer. An interesting edge case was noted during this process: for an answer like "treasure_chest," Zephyr consistently outputs "treasure_", which I deemed correct for my evaluation.

4.1 Datasets

4.1.1 SocialIQA

The socialIQA data Set downloaded came in JSON format, with five keys for each object. (context, question, answerA, answerB, answerC) After the data had been read, wsc from template for creating training data for Flab-T5 is used to reconstruct the questions, as shown below. The answer returned by the model is always A, B or C.

Context: Alex, the on call nurse, eased their pain with medicine.

Question: How would you describe Alex?

answerA: a lazy worker

answerB: hates people

answerC: a caring person

Figure 4.1: Example of socialIQA data, bold text is the key, and plain text is value

```
input_data = data["context"] + "\n\n" +  
data["question"] + "\n\n" +  
"A: " + data["answerA"] + "\nB: " + data["answerB"]  
+  
"\nC: " + data["answerC"]
```

Figure 4.2: Example of how socialIQA datais formatted

The result is quite close to the result on the leaderboard. As the model on the leaderboard is Flan-T5 XXL, the performance gap is understandable. The result is very close to human [27], with the smaller model, Flan-T5, with only 250M parameters, outperforming GPT3-daVinci (56% with 35 examples). This could be because T5 is better at ToM tasks, or the dataset itself is not challenging.

4.1.2 ToMi

The data set is in text files, and each sentence is in its line and indexed. The last sentence of one question always ends with a question mark, and the answer is after the question mark. natural question template is used, and the input to the model is a string with sentences concatenated and index removed. The answer returned by the model is always a single word.

- 1 Oliver entered the porch.
- 2 Owen entered the porch.
- 3 Abigail entered the porch.
- 4 The broccoli is in the bucket.
- 5 Abigail exited the porch.
- 6 Owen exited the porch.
- 7 Owen hates the peach
- 8 Oliver moved the broccoli to the pantry.

Where was the broccoli at the beginning? bucket 1

Figure 4.3: Example of ToMi data

4.2 Models

4.2.1 Flan-T5-base

This model is the smallest used for this project, with only 250M parameters [4]. However, despite its small size, it actually performs better than other, much larger models. The high performance may suggest that the model is 'cheating,' possibly because the datasets are included in the model's original training set, or the model has been specifically fine-tuned on these datasets.

It is also the easiest model to use for inference, with API calls from Hugging Face. With robust support for batching using the `generate()` method, the inference

speed is very fast, allowing both SocialIQA and ToMi to be evaluated in just a few hours.

4.2.2 Llama3-8B

Llama3 is a brand-new model that was released in April 2024. It comes in two versions, 8B and 70B. For this project, we used the 8B version because of its smaller size and faster running time. Surprisingly, it performs exceptionally well compared to the Mistral 7B, allowing for more tests to be conducted on the model.

There are two versions of Llama3 8B: Instruct and non-Instruct. For our test questions, which are in the form of QA pairs, we used Llama3 8B Instruct. The model uses a chat template for inputs, so there is a system role. Compared to Zephyr, the input to the system input is very reliable; it precisely follows the system role's instructions. For SocialIQA, the system input is '*Always answer with A, B, or C*', and the model output is very consistent; the output is always a single label.

For ToMi, the system input is '*Always answer with only one word*', and the output is also very consistent, outputting only the label word.

We also tested the Chain of Thought (CoT) performance with this model, due to its fast inference speed. From Table 4.2, we see that there is only limited improvement with CoT. This finding shows that CoT may not be suitable for all use cases."

Listing 4.1: Chat template for Llama3 8B

```
1 [
2     {'role': 'system', 'content': system_input, },
3     {'role': 'user', 'content': user_input}
4 ]
```

One interesting observation is that the model appears to be more cautious with illegal or sexual content. This problem is not encountered with all other models. It is likely because this model is the only one that is widely commercially used, making it more restricted and mature in these areas. When the context contains topics related to crime or sexual content, it will respond with the phrase '*I cannot provide information or guidance on illegal or harmful activities. Can I help you with something else?*' Interestingly, the same behaviour is observed when using ChatGPT, when attempting to generate results for Table 4.1. However, ChatGPT did not refuse to answer, but issued a warning message: '*This content may violate our usage policies. Did we get it wrong? Please tell us by giving this response a thumbs*

down.' This suggests that our explanation for why Llama3 refused to answer is correct: these models are commercially used and are manually set to be cautious with dangerous topics.

We classify these responses as 'Failed to answer' in Table 4.2. There are only a few cases with this problem, so it does not significantly affect the overall accuracy.

Example Context
<ul style="list-style-type: none"> • Carson pulled lint out of Tracy's pocket while they were trying to steal something. What does Carson need to do before this? • Alex grabbed his girlfriend's breast when they were having sex for the first time. What will happen to his girlfriend?
Model Output
I cannot provide information or guidance on illegal or harmful activities. Can I help you with something else?

Table 4.1: Question and Answer Context.

4.3 The Mistral Family

The rest of the models are all from the Mistral family, and they all accept input in chat format, similar to Llama 3. With Llama 3, the model behaves exactly as instructed by the system input, but with models from the Mistral family, the story is different. Mistral family models tend to ignore the instructions from the system input and produce very unstructured outputs. All models' outputs exhibit the composite answer problem.

4.3.1 Mistral-7b

Mistral-7b is developed by the Mistral.ai team and outperforms larger models like Llama 2 13B across all benchmarks. Its compact size allows it to operate with limited resources, and its impressive performance makes it worthwhile to evaluate its ToM capabilities.

Although it is smaller compared to models like Llama 2 13B, its file size is about 13GB, which complicates its usage via the HuggingFace API. In fact, when attempting to run the model on a DCS GPU with 24GB of RAM, it fails and reports an 'out of memory' error.

Thus, we tested the copybarahermes-2.5-mistral-7b.Q4_K_M.gguf model, which is 4.37 GB in size and has a maximum RAM capacity of 6.87 GB. This model, a

quantized version from TheBloke, offers a balance of size and quality and comes recommended.

The model is small enough to run on DCS's server, but it still takes a considerable amount of time. On average, it takes about 2 minutes (126813.84 ms) to process a single question. Due to the slow inference speed, only 80% of each dataset has been evaluated.

Inference with this model requires the use of the function 'create_chat_completion'. This is highly convenient, as both the question and the answer can be inputted into a list, fed into the function, and then used to generate the desired model input.

the list:

```
messages = [  
    {"role": "system", "content": system_input},  
    {"role": "user", "content": question},  
]
```

The Mixtral 8x7B model is a Sparse Mixture of Experts model (SMoE) [11]. It boasts significant advantages in terms of performance and inference speed, being six times faster than the Llama 2 70B model and outperforming it on most benchmarks. It is competitive with GPT-3.5, matching and sometimes surpassing it on several standard benchmarks.

This model is the largest encountered so far. With its quantized version, it is 24.63GB and barely fits on the Avon server's Quadro RTX 6000 GPU, which has 24GB of RAM. Although it is fast compared to a 70B model, it still takes about 9 minutes (533888.75 ms) on average to process a single question. Due to the slow inference speed, only 50% of each dataset has been evaluated.

The model is inferences with the following code, where [INST] mark the starting and ending point of the prompt to the model.

Listing 4.2: Example Code

```
output = llm(  
    f '[INST]{prompt}[/INST]', # Prompt  
    max_tokens=512, # Generate up to 512 tokens  
    stop=[  
        '</s>'  
    ], # Example stop token  
    echo=True, # Whether to echo the prompt  
)
```

4.3.2 zephyr-7b-beta

Zephyr [29] is a chat model that requires chat template inputs. The difference between chat models and language models is that rather than expose a “text in, text out” format, chat models expose an interface where “chat messages” are the inputs and outputs.

Huggingface provides chat templates as a feature, which is part of the tokenizer. “assistant”, “system”, and “user” are roles in the template, and their content is message texts.

The “system” role is used to give directives to the model about how it should behave in the subsequent conversation. This allows the chatbot to set up its behaviour before interacting with the user. The “user” role represents the messages the user sends in the conversation. In the context of a chat template, messages from the user are formatted and tokenized in a way that the model can understand and respond to. The user role is crucial as it drives the conversation and prompts responses from the assistant. The “assistance” role corresponds to the responses generated by the model. The assistant’s responses are based on the user’s input and previous system direction. The assistant role is designed to interact with the user, provide answers and information, or engage in conversation as programmed.

Zephyr was chosen over Mistral due to its straightforward guidelines, while Mistral lacks sample code on its HuggingFace page, and code from online tutorials fails to work on Google Colab. One issue with the `pipeline()` function is that it does not support batching perfectly; the documentation states that batching can sometimes even reduce performance. Therefore, all tasks are run individually on the Avon server. This takes significantly longer than running T5, and resources are not always available.

As mentioned in Section 3.1.5, when dealing with double inference, we may choose a faster model for the second inference. In Table 4.2, ‘Double-inference (Zephyr)’ indicates that the second model is Zephyr, and ‘Double-inference (Flan-T5)’ means the second model is Flan-T5. Data from Table 4.2 shows that if the second model is Flan-T5, the accuracy increases dramatically. This suggests that Flan-T5 improves the results, as it is effective with both datasets. Therefore, the best practice for double inference is to use the original model for the second inference. Using another model may increase or decrease performance depending on the second model’s performance with the dataset.

4.4 Experiments

Model	Dataset	Prompting Strategy	Accuracy	Failed to Answer	Accuracy w/o Failed Cases
Flan-T5-base	SocialIQA	Standard	81%	-	-
	ToMi	Standard	71%	-	-
Zephyr-7b-beta	SocialIQA	Standard	35%	15109	63%
		Overprompt	20%	7760	-
		Overprompt (PR)	24%	5951	-
		JSON Format	66%	-	68%
		D-I	58%	-	-
		CoT + D-I	83%	-	-
	ToMi	Standard	57%	60	57%
Mistral-7b	SocialIQA	Standard	78%	-	-
	ToMi	Standard	76%	-	-
Mixtral-7b*8	SocialIQA	Standard	88%	-	-
	ToMi	Standard	72%	-	-
Llama3-8B	SocialIQA	Standard	74%	172	74%
		CoT + D-I	75%	257	76%
	ToMi	Standard	67%	-	-
		CoT + D-I	66%	-	-

Table 4.2: Accuracy of different Large Language Models across various datasets and prompting strategies. ‘PR’ indicates ‘Phrase Removed,’ ‘CoT’ refers to ‘Chain-of-Thought,’ and ‘D-I’ denotes ‘Double Inference.’

4.4.1 Experimental Results

We proceed with a detailed discussion of the experimental results on the adopted datasets. Table 4.2 presents the results of various models across different datasets and prompting strategies, revealing interesting insights into their performance. Flan-T5-base achieved 81% and 71% accuracy on SocialIQA and ToMi, respectively, highlighting the model’s baseline capabilities. However, when it comes to Zephyr-7b-beta, there’s a significant variation in performance based on the strategy applied. Standard prompting on SocialIQA yields 35% accuracy, which rises dramatically to 83% with the addition of Chain of Thought (CoT) and Double-Inference, emphasizing the substantial impact of combining these methods. Interestingly, Overprompt shows a markedly lower performance on SocialIQA, at 20% and 24%, indicating it might not be as effective as the CoT and Double-Inference approach.

Mistral-7b performs well on both datasets, with 78% and 76% accuracy on

SocialIQA and ToMi, respectively, while Mixtral-7b*8 outperforms it on SocialIQA with 88% accuracy, showcasing Mixtral’s enhanced capacity for understanding social scenarios. Llama3-8B’s performance is notable, too, especially with CoT and Double-Inference, reaching 75% accuracy on SocialIQA, which demonstrates the positive effect of these strategies on the model’s reasoning abilities.

Overall, the results suggest that prompt engineering strategies like CoT and Double-Inference significantly improve model performance, while Overprompt may not be as effective. The variability in performance across different models and prompting strategies emphasizes the importance of adapting strategies to the specific model and task for optimal results.

4.4.2 Qualitative Analysis

We continue our analysis providing some of the most interesting insights into the output generated by different models over different datasets.

Results on the SocialIQA

Many instances of “Failed to Answer” resemble the example below, where the model selects the fourth option, “Unknown”. This indicates the model’s uncertainty about the correct response, and the rationale provided seems justifiable. Upon scrutinizing the question, it becomes apparent that the quality of the answers is subpar. The most accurate answer is A, which is superior. After evaluating all the options, A emerges as the closest to being correct. However, in scenarios like “make Skylar’s son better”, option A appears more appropriate. Also, the model does conclude that the answer is A, but its position is quite random, unsuitable for automatic evaluation, and more needs to be done on further work.

<|system|>

Answer only A, B or C.</s>.

<|user|>

Addison gave a hug to Skylar’s son when they were feeling down. Why did Addison do this? A: better B: wrong C: keep hugging the son</s>

<|assistant |>

A is not a valid answer as it does not provide a reason for Addison’s actions. B is not a valid answer as it suggests that Addison did something wrong. C is not a valid answer as it is not a plausible reason for Addison’s actions.

Addison likely hugged Skylar’s son because she wanted to comfort and support him when he was feeling down. Comforting someone through a hug is a common way to show empathy and care. Therefore, the correct answer is A, but we need to provide a reason for Addison’s actions, which would be: Addison hugged Skylar’s son to offer him comfort and support during a difficult time.

Figure 4.4: Example of zephyr output.

ToMi

Regarding the limited number of tokens, the complete answer is not accessible. However, the frequency of failure instances is minimal, making them more like exceptional cases. The question format mirrors that of T5 and adopts a user role perspective. The system’s role is rigidly set to “Answer in one word,” as illustrated in the example below.

```
<|system|>  
Answer in one word.</s>
```

```
<|user|>  
...</s>
```

```
<|assistant|>  
...
```

Zephyr on Transparent Access

The example below illustrates Zephyr operating with transparent access. The response is well-aligned with the original question, which did not specify transparent access. This indicates that the model is either insensitive to, or unaware of, the impact that a transparent container would have on the scenario.

```
<|system|>  
Answer the question.</s>
```

```
<|user|>  
Here is a bag filled with popcorn. There is no chocolate in the bag. The  
bag is made of transparent plastic, so you can see what is inside. Yet, the
```

label on the bag says 'chocolate' and not 'popcorn'. Sam finds the bag. She had never seen the bag before. Sam reads the label. She believes that the bag is full of

`<|assistant|>`

Chocolate, despite the fact that the bag is actually filled with popcorn and the label is misleading. This is known as the “curse of knowledge” or the “mislabel effect,” where prior knowledge or experience can lead to incorrect assumptions or beliefs. In this case, Sam’s prior knowledge that the bag should contain chocolate, based on the label, overrides her visual perception of the popcorn inside.

Results of the Structured Prompt - JSON Format

This method was introduced to tackle the composite answer problem, making it ideal to accompany models from the Mistral family, as these models inherently exhibit the composite answer problem. However, only Zephyr was tested with this prompting strategy, due to the slow inference speed of Mistral 7B and Mixtral 8*7B.

This method is quite efficient, resulting in the model behaving much better. Unlike normal prompting strategies, the label is now consistently at the front of the output. However, the model still tends to make minor changes to each output. Sometimes the label is in lowercase and sometimes uppercase; at times, the label is wrapped by double quotes; occasionally, it provides the label’s answer (e.g., if a label is 'answerA: Hello.', then the label’s answer is 'Hello') instead of the label itself; and sometimes it responds with 'answerA' instead of 'A'. These issues show that the model is quite difficult to control. However, they can be easily fixed with string operations, making them easier to solve than the composite answer problem.

Nonetheless, the issue with poor questions from the dataset persists. Sometimes the model generates something similar to the following quote, which requests more information. The model is uncertain about the question and refuses to give a label. This type of behaviour is classified as 'Failed to answer' in Table 4.2. There are not many cases of failure to answer, but even with the failed cases removed, there is still a noticeable increase in accuracy. This further suggests that there are noticeable flaws in the dataset.

`<|user|>`Can you suggest some strategies Sydney can use to deal with Riley’s teasing? Based on the given context, what do you think would be most effective?

Chapter 5

Project Management

5.1 Running Time

One of the biggest issues in conducting experiments was running the models correctly on the servers. While ChatGPT can respond in seconds, this project was constrained by limited resources. The majority of the experiments were run on the Avon server, equipped with a Quadro RTX 6000 (24GB) GPU, and some on DCS's Kudu server, with an Nvidia A10 (24GB) GPU. Although these are powerful GPUs, the models used in this project still ran slowly, especially the Mistral family models, which were particularly slow. There may be some custom settings required, but no potential solutions were identified by the time this report was prepared. With models like Zephyr or Flan T5, even when called via API, they can process a whole dataset within a day. For example, Zephyr can complete SocialQA in 23 hours. However, even with the quantized version, Mixtral 8x7B takes 2 days to process 200 to 300 questions on Quadro. With SocialQA and Mistral 7B on A10, it takes 2 days to complete 1300 questions on average. Both servers have a time limit of 2 days. With this speed, the experiments cannot be completed fully, so only part of the dataset was tested.

Additionally, during the early and middle stages of the project, there was a consistent issue that blocked inference with Mixtral 8x7B. The Avon server uses the SLURM system to manage jobs. When submitting a Mixtral 8x7B inference job via the 'sbatch' command, the model took a total time of 43,535,410.83 ms to process one question, which is about 12 hours. This problem was not resolved until the end of term 2. It was discovered that using 'salloc', which acquires a computation node for interaction with the current terminal, significantly reduced the time taken to about 554,814.09 ms, or approximately 9 minutes. Experiments were then conducted in

this manner. A lot of time was wasted trying to find a solution to the long running times.

One problem with Zephyr is calling APIs is the lack of support for batching. Although the method technically supports batching, it actually results in a slowdown rather than a speedup.

Interestingly, Llama3 8B, which is 1B larger in size, is significantly faster than Mistral 7B. It loads in 22.63 seconds with a single SocialIQA question and processes 34 questions per batch in 20.09 seconds for SocialIQA inference, and 18.75 seconds per batch for ToMi. This might be due to the fact that Llama is actually commercially used, and inference speed is crucial for regular use. Thanks to the fast inference speed, the Chain of Thought (CoT) can be tested, along with zero-shot prompting.

5.2 JSON Format

The JSON format prompting strategy took the longest to function correctly. It was initially proposed at the start of term two but only started working towards the end of the project. Part of the reason was poor code file management, but the main issue was a bug in the function that packs the dataset's questions into JSON format. Zephyr is a model that requires a chat format, necessitating a specific format for the input. When the function was first written, this specific format was defined incorrectly, but this mistake was not discovered until the end of the project.

Before this bug was found, the model was generating random output. This was originally thought to be an inefficiency of the JSON format prompting strategy. It was not until the end of the project that the original code was reviewed again.

5.3 TimeTable

Throughout the project, my approach to project management was largely influenced by my supervisor's guidance on the direction of the work. In term one, we established a routine of holding two weekly meetings, a schedule that we maintained from Week 5 to Week 8. This management style functioned similarly to a module, emphasizing the gradual building of knowledge and skills. Accordingly, the rate at which work was produced was closely tied to my learning pace.

However, during term one Week 8, I encountered a significant setback when I developed back pain, which severely limited my ability to sit and work, resulting in a noticeable decrease in productivity. I requested an extension for the project,

and consequently, the deadline for submitting my progress report was extended to Monday of term one week 10.

Term two was the period when most experiments were conducted. It took time to become accustomed to the server's management system. One difficulty was setting up the models; the process was quite unfamiliar, whether by calling an API or running locally. It took weeks to fully understand how to use Zephyr via Hugging Face's API. However, this experience was quickly leveraged for Llama3, a model that was used towards the end of the project. It took only a day to become familiar with the model and another day to start all the experiments—operations that had taken days when setting up Mistral and Zephyr.

It was not until midway through term two that I stopped reading new papers. Before this, I was quite unconfident about what to do and therefore spent a lot of time reading and searching for new papers. It wasn't until two weeks before the presentation that I fully started experimenting and stopped reading new papers. Each paper took several hours to finish, and additional time was needed for literature reviews, taking about a whole day. At that time, I realized that I would have limited results to show and stopped worrying that I might need to read more. It turns out that the worry was unnecessary, as many of the papers I read did not contribute much to my project, and some were not useful at all. This was due to the fact that the information from these papers was not connected with the content from the rest of the project. This caused quite a bit of delay in progress.

Several events caused unnecessary time wastage for this project. One significant issue was file management. Three machines were used for this project: my own laptop, DCS's machine, and the Avon server, each with its own storage system. In Term One, only the Avon server and my laptop were used, so all the code was kept on the Avon server, and written files and papers were stored locally. In Term Two, when DCS's server was introduced, confusion quickly arose as code files were not synced. This problem was largely resolved when GitHub was employed to track everything. However, having never used GitHub or any Git system before, it took some time to get everything operational. This initially caused delays in the project, but having a synced environment ultimately helped save time and energy in the long run.

Additionally, due to poor file management on my local machine, some files were disorganized and poorly labeled. This caused confusion and wasted time during report meetings with my supervisor during the term. The issue was resolved with better organization of files and improved labeling.

Time	Work
T1 W1-W2	Read several papers and finish specification document
T1 W3-W5	Read more papers and set up directions for the term, try to recreate results from the papers
T1 W5-W7	Implement a working code for some models and data sets, meeting with supervisor twice a week to keep on track
T1 W7-W8	Draft progress report and continue W5-W7's work
T1 W8-W9	Finish progress report
T1 W10 Monday	Progress report
T2 W1-W2	Read several papers, trying to implement overprompt.
T2 W3-W5	Tried several ways to extract answers. E.g., overprompt, Json format prompt, cosine similarity. Using Avon an Dcs's server to run the models.
T2 W5-W8	Tried Mistral-7b-GGUF and Mistral-7b*8-GGUF. Modified Json format prompt and overprompt trying to get better performance.
T2 W8-W9	Draft presentation, working on demo system.
T2 W10 Monday	Presentation

Table 5.1: Work schedule.

Time	Work
Christmas	test models on the framework read more about the models
T2 W1-W2	Draft the first half of final report
T2 W2-W6	Continue testing, summarise the successful and failed cases
T2 W6-W8	Draft presentation and continue work from previous work
T2 W8-W9	Finish presentation
T2 W9-W10	Presentation
Six weeks from presentation	fix all bugs finish the final report, and send a draft in week 4.

Table 5.2: original Term 2 timetable.

Chapter 6

Conclusion

6.1 Contribution

For this project, existing work on ToM was revisited, and some solutions were proposed to tackle problems arising from some of the existing works.

In the case of SocialIQA, it was discovered that some of the questions and answers were poorly written, causing models to become confused and give either correct or incorrect labels. With FanToM, it was found that Cosine similarity might not be consistent or accurate in determining whether the model's output is correct. The JSON format and double inference were used to tackle the composite answer problem. Both methods have proven to be effective in solving this issue.

Methods to increase ToM abilities have also been tested in this project. CoT has been used, and improvements in the model's ToM performance have been observed. However, it has also been shown that the improvement trend is not consistent; it may be model or dataset dependent.

OverPrompt was used to speed up inference time and test for a potential increase in ToM ability by utilizing the models' in-context learning ability. However, it has been shown that while it can significantly boost inference speed, performance also significantly dropped.

6.2 Limitations

There are many ToM datasets available today. However, due to the complex nature of different ToM datasets, only a few that were encountered in the early stages of the project are selected and considered for my project. These datasets are simpler compared to relatively newer datasets, as they were introduced quite some time

ago. For instance, SocialIQA was introduced on September 9, 2019, and ToMi on November 3, 2019. In comparison, FanToM was introduced on October 24, 2023, and OpenToM on February 14, 2024. The problems of composite answers or the quality of these datasets remain unknown. Furthermore, data from just these two older datasets may not be as reliable today.

The models used for this project are relatively small due to limited computational resources available. With larger models such as Llama3 70B, or less quantized versions of Mistral and Mixtral, the performance should be better.

6.3 Future Work

As the project only investigated a few datasets and models, future work could test more datasets with various models, particularly those using quantization techniques. The project operates in a setting with limited computational resources, and it is crucial that further research also explores the Theory of Mind (ToM) abilities of large language models under these constraints. This is especially important as not everyone has access to massive computing power. From this project, double inference is used to address the composite answer problem. With further work, we can explore alternative approaches to extract answers using Schema Enforcer. It has been shown that simple prompting strategies, such as the CoT, can enhance models' ToM abilities. However, this may be model or type of dataset dependent. Therefore, it is worthwhile to conduct large-scale experiments using the CoT framework. Evaluating CoT under diverse settings will provide insights into how it behaves. With modern and more powerful models, problems in the dataset can be identified. In the future, we may explore whether these models can detect various problems across different datasets, using either more advanced future models or existing ones. With additional ToM datasets, it might be worthwhile to add a third option (either 'yes' or 'no') for binary choice questions. We have observed that sometimes the model may be unable to decide and fails to answer. This addition would serve as a safeguard for datasets, testing whether there are flaws in the dataset. Specifically, if many models cannot decide on the same question, then this question from the dataset may be ambiguous.

Appendix A

Cosine Similarity Check

This section show the questions that cosine similarity got wrong, with output from Zephyr-7b-Beta and the FanToM[13] dataset. There are only three attributes shown, the output from Zephyr, the label list and the tensor. The first element of the tensor value represent the first element in the label list, and the second tensor value represent the second label list element. The FanToM question and Context are too long, and it is enough to justify if the tensor is correctly classified the model output by the output and the label. The quoates below are some examples model all-roberta-large-v1(1355MB), got wrong:

Zephyr's output: 'The conversation did not specifically mention a special challenge that Elena faced in parenting. The discussion focused on balancing challenges, emotional connections, and the rewards of parenthood in general. Elena shared her perspective on creating a safe space for her son and the importance of empathy, understanding, and open communication for his emotional growth. However, there was no mention of a specific challenge she faced in this regard.'

Label List: ["Elena's special challenge in parenting was maintaining a balance between wanting to protect and nurture her child but also letting him learn things by himself.", 'The conversation does not provide specific information on a special challenge that Elena faced in parenting.']

Encoder's Output: tensor([[0.8149, 0.7590]])

Zephyr's output: "The text doesn't provide specific life lessons that Alayna took away from her upbringing. The conversation between

Alayna, Jaden, and Joaquin revolves around sharing their unique experiences growing up, but it doesn't delve into the specific lessons they learned from their families. However, Alayna did mention that having two ways of life without expectations made her feel free, which could suggest that she values independence and individuality. Additionally, she expressed gratitude for the introspective talk, which could indicate that she finds value in reflecting on her past and learning from her experiences."

Label List: ['Alayna took away lessons about resilience and coping with change from her upbringing.', 'The text does not provide information on what life lessons Alayna took away from her upbringing.']

Encoder's Output: tensor([[0.6592, 0.6432]])

Zephyr's output: "Alana and Drake discussed their favorite travel destinations, but Zachery joined the conversation and shared his favorite travel destination, which was Japan. It is not explicitly stated which destinations were Alana and Drake's favorites."

Label List: ['Zachery does not know about the favorite travel destinations that Alana and Drake discussed because he was not involved in the conversation when this was discussed.', 'Zachery believes that the favorite travel destinations Alana and Drake discussed were Bali, Indonesia and Greece.']

Encoder's Output: tensor([[0.7525, 0.8191]])

Below are the examples that all-MiniLM-L6-v2(80MB) got wrong:

Zephyr's output: 'Gianna does not explicitly state whose pets Javier and Sara were discussing. The conversation seems to be a general discussion about training and bonding with pets, and each person shares their own experiences with their own pets.'

Label List: ["Gianna believes that Javier and Sara were discussing Bruno and Chirpy's pets.", 'Gianna believes that Javier and Sara were discussing their own pets, Bruno and Snowflake.']

Encoder's Output: tensor([[0.8205, 0.7511]])

Zephyr's output: 'The conversation in the text does not directly address

Veronica's belief about how Fatima overcomes the challenge of dealing with different personalities at work. Veronica only joins the conversation towards the end, and her contribution is limited to sharing her own experience of setting boundaries and learning to say "no" when work expectations become unrealistic. The conversation does not delve into how Fatima specifically handles dealing with different personalities at work.'

Label List: ['Veronica does not know or is unaware about how Fatima overcomes the challenge of dealing with different personalities at work because she was not involved in the conversation when this was discussed.', 'Veronica believes that Fatima overcomes the challenge of dealing with different personalities at work by observing, learning, and adopting certain assertive communication strategies.']

Encoder's Output: tensor([[0.7615, 0.8686]])

Zephyr's output: "Jaiden does not express a belief about Veronica's belief regarding how Fatima overcomes the challenge of dealing with different personalities at work in this conversation. Veronica shares her belief that Fatima is patient and understanding when dealing with different personalities at work, but Jaiden does not respond to this statement with his own belief. Instead, he moves on to another topic in the conversation."

Label List: ['Jaiden believes that Veronica does not know how Fatima overcomes the challenge of dealing with different personalities at work because Veronica was not involved in the conversation when this was discussed.', 'Jaiden believes that Veronica thinks Fatima overcomes the challenge of dealing with different personalities at work by observing, learning, and adopting certain assertive communication strategies.']

Encoder's Output: tensor([[0.7682, 0.8510]])

These are the common error from both all-MiniLM-L6-v2 and all-roberta-large-v1:

Zephyr's output: 'Alayna believes that Joaquin thinks the life lessons she took away from her upbringing are unique and interesting due to the contrast between her privileged lifestyle and her parents' divorce.'

However, she also acknowledges that Joaquín's own blend of cultures has added richness to his upbringing and shaped him in his own way. Alayna's statement, "It seems like despite the varied backgrounds, we all have unique aspects in our upbringing that have shaped us in some way," suggests that she believes Joaquín recognizes the value and significance of each person's individual experiences.'

Label List: ['Joaquín is unaware of what Alayna believes about the life lessons she took away from her upbringing, as he was not involved in the conversation when this was discussed.', 'Joaquín believes that Alayna took away lessons about resilience and coping with change from her upbringing.']

all-roberta-large-v1's Output: tensor([[0.6850, 0.6846]])

all-MiniLM-L6-v2's Output: tensor([[0.8167, 0.8170]])

Zephyr's output: "The discussion did not specifically focus on responsibilities as parents, but some responsibilities were mentioned in passing. For example, Kobe expressed his desire to continue learning from his fellow parents, implying a responsibility to continue growing and improving as a parent. Elena also mentioned the importance of exchanging stories, lessons, and laughter, which could be seen as a responsibility to support and learn from other parents. Additionally, Silas's story about his son's request to bring a lion home highlighted the responsibility to explain practicalities to children in a way they can understand. However, the discussion primarily focused on sharing experiences and insights about parenting, rather than delving into specific responsibilities."

Label List: ["The responsibilities discussed by the group included fostering emotional connections with their children and creating a safe space for open communication. They also talked about letting their children learn by themselves while still providing the necessary protection and nurturing, as well as putting in the effort to avoid comparing their children's development with others'. Balancing career responsibilities with being there for their children and dealing with parenting challenges were also discussed.", 'The group did not specifically discuss any parenting responsibilities. They were sharing personal experiences and memorable stories about their children.']

all-roberta-large-v1's Output: tensor([[0.6368, 0.5409]])

all-MiniLM-L6-v2's Output: tensor([[0.6489, 0.6670]])

Zephyr's output: "The text does not explicitly state who predominantly raised Alayna after her parents separated. It only mentions that Alayna's parents separated when she was young, and she spent time with both her mom and dad without expectations. However, it can be inferred that Alayna's mom may have predominantly raised her as she mentioned feeling free and without expectations during her time with her mom. This is not confirmed, and further information would be needed to make a definitive statement."

Label List: ['Alayna was predominantly raised by her mom and her older sister after her parents separated.', 'The text does not provide information on who predominantly raised Alayna after her parents separated.']

all-roberta-large-v1's Output: tensor([[0.8799, 0.7252]])

all-MiniLM-L6-v2's Output: tensor([[0.8036, 0.8140]])

Appendix B

GPT4 on Altered Tasks

GPT4 on the altered Unexpected Contents Task and Unexpected Transfer Task by Ullman[30]. In the paper, GPT-3.5 got all the questions wrong. But with GPT4, there are 2 out of total 16 questions that are answered correctly. The correct cases are coloured red. The fact that GPT4 answered two correctly does not show that GPT4 is more capable at ToM, as under the same context, both correct cases got the other question wrong.

Transparent Access with Unexpected Contents Task

Context:Here is a bag filled with popcorn. There is no chocolate in the bag. **The bag is made of transparent plastic, so you can see what is inside.** Yet, the label on the bag says 'chocolate' and not 'popcorn.' Sam finds the bag. She had never seen the bag before. Sam reads the label

First Prompt:She believes that the bag is full of [FILL]

Model Output:Based on the scenario described, where Sam reads the label on the bag that says "chocolate" and has not seen the bag before, she would likely believe that the bag is full of **chocolate**. Even though the actual contents are popcorn, her belief would be based on the label she reads.

Second Prompt:She is delighted that she has found this bag. She loves eating [FILL]

Model Output:Based on the information provided, Sam reads the label on the bag, which says "chocolate," and she is delighted upon finding it because she believes it contains chocolate. Therefore, she loves eating **chocolate**. Her delight suggests that she is responding to what she reads on the label, assuming it's accurate, despite the visible contents being popcorn.

Uninformative Label with Unexpected Contents Task

Context: Here is a bag filled with popcorn. There is no chocolate in the bag. Yet, the label on the bag says “chocolate” and not “popcorn.” Sam finds the bag. She had never seen the bag before. She cannot see what is inside the bag. **Sam cannot read.** Sam **looks** at the label.

First Prompt: She believes that the bag is full of [FILL]

Model Output: Sam believes that the bag is full of **chocolate**.

Second Prompt: She is delighted to have found this bag. She loves eating [FILL]

Model Output: **chocolate.**

Trustworthy Testimony with Unexpected Contents Task

Context: Here is a bag filled with popcorn. There is no chocolate in the bag. The label on the bag says “chocolate”, rather than “popcorn.” **Before coming into the room, Sam’s friend told her ‘the bag in the room has popcorn in it, ignore the label’.** Sam believes her friend. Sam finds the bag. She had never seen the bag before. She cannot see what is inside the bag. Sam reads the label, **which says the bag has chocolate in it.**

First Prompt: She believes that the bag is full of [FILL]

Model Output: Sam believes that the bag is full of **popcorn**.

Second Prompt: She is delighted to have found this bag. She loves eating [FILL]

Model Output: **chocolate.**

Treachery of Late Labels with Unexpected Contents Task

Context: Sam fills a bag with popcorn and closes it. There is no chocolate in the bag. Sam writes a label and puts it on the bag. Sam looks at the bag. She cannot see what is inside the bag. Sam reads the label. The label says the bag has chocolate in it.

First Prompt: She believes that the bag is full of [FILL]

Model Output: **popcorn**

Second Prompt: She is delighted to have found this bag. She loves eating [FILL]

Model Output: **chocolate.**

Transparent Access with Unexpected Transfer Task

Context:In the room there are John, Mark, a cat, a **transparent plastic** box, and a **glass chest**. John takes the cat and puts it in the **chest**. He leaves the room and goes to school. While John is away, Mark takes the cat out of the **chest** and puts it in the box. Mark leaves the room and goes to work. John comes back from school and enters the room. He doesn't know what happened in the room when he was away.

First Prompt:John thinks that the cat is in the [FILL]

Model Output:John thinks that the cat is in the **chest**.

Second Prompt:John will look for the cat in the [FILL]

Model Output:John will look for the cat in the **chest**.

Relationship Change with Unexpected Transfer Task

Context:In the room there are John, Mark, a cat, a box, and a basket. John takes the cat and puts it on the basket. He leaves the room and goes to school. While John is away, Mark takes the cat **off the** basket and puts it on the box. Mark leaves the room and goes to work. John comes back from school and enters the room. **John looks around the room**. He doesn't know what happened in the room when he was away.

First Prompt:John thinks that the cat is in the [FILL]

Model Output:John thinks that the cat is on the **basket**.

Second Prompt:John will look for the cat on the [FILL]

Model Output:John will look for the cat on the **basket**.

Trusted Communication with Unexpected Transfer Task

Context:In the room there are John, Mark, a cat, a box, and a basket. John takes the cat and puts it in the basket. He leaves the room and goes to school. **Mark calls John to tell him he is going to move the cat to the box. John believes him**. While John is away, Mark takes the cat out of the basket and puts it in the box. Mark leaves the room and goes to work. John comes back from school and enters the room. He doesn't know what happened in the room when he was away.

First Prompt:John thinks that the cat is in the [FILL]

Model Output:John thinks that the cat is in the **basket**.

Second Prompt:John will look for the cat on the [FILL]

Model Output:John will look for the cat in the **basket**.

Querying the Mental States of the Additional Person with Unexpected Transfer Task

Context:In the room there are John, Mark, a cat, a box, and a basket. John takes the cat and puts it in the basket. He leaves the room and goes to school. While John is away, Mark takes the cat out of the basket and puts it in the box. Mark leaves the room and goes to work. John **and Mark** come back and enter the room. **They don't** know what happened in the room when **they** were away.

First Prompt:John thinks that the cat is in the [FILL]

Model Output:John thinks that the cat is in the **basket**.

Second Prompt:John will look for the cat on the [FILL]

Model Output:John will look for the cat in the **basket**.

Bibliography

- [1] URL <https://llama.meta.com/llama3/>.
- [2] Abid Ali Awan. Comprehensive guide to zephyr-7b: Features, usage, and fine-tuning, Dec 2023. URL <https://www.datacamp.com/tutorial/zephyr-7b-tutorial>.
- [3] Lindsay J. Byom and Bilge Mutlu. Theory of mind: mechanisms, methods, and new directions. *Frontiers in Human Neuroscience*, 7:413, 2013. doi: 10.3389/fnhum.2013.00413.
- [4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.
- [5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.
- [6] David Dodell-Feder, Sarah Hope Lincoln, Joseph P. Coulson, and Christine I. Hooker. Using fiction to assess mental state understanding: A new task for assessing theory of mind in adults. *PLOS ONE*, 8(11):null, 11 2013. doi: 10.1371/journal.pone.0081279. URL <https://doi.org/10.1371/journal.pone.0081279>.
- [7] FG Happé. An advanced test of theory of mind: understanding of story characters’ thoughts and feelings by able autistic, mentally handicapped, and

normal children and adults. *Journal of autism and developmental disorders*, 24 (2):129—154, April 1994. ISSN 0162-3257. doi: 10.1007/bf02172093. URL <https://doi.org/10.1007/BF02172093>.

- [8] Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. A fine-grained comparison of pragmatic language understanding in humans and language models. *arXiv preprint arXiv:2212.06801*, 2022.
- [9] Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. A fine-grained comparison of pragmatic language understanding in humans and language models, 2023.
- [10] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b, 2023.
- [11] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th  ophile Gervet, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mixtral of experts, 2024.
- [12] Cameron Robert Jones, Sean Trott, and Ben Bergen. EPITOME: Experimental protocol inventory for theory of mind evaluation. In *First Workshop on Theory of Mind in Communicating Agents*, 2023. URL <https://openreview.net/forum?id=e5Yky8Fnvj>.
- [13] Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. FANTom: A benchmark for stress-testing machine theory of mind in interactions. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL <https://openreview.net/forum?id=5TEfD2GBUc>.
- [14] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

- [15] Michal Kosinski. Theory of mind may have spontaneously emerged in large language models. *ArXiv*, abs/2302.02083, 2023. URL <https://api.semanticscholar.org/CorpusID:263890629>.
- [16] Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 5872–5877, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1598. URL <https://aclanthology.org/D19-1598>.
- [17] Jiazheng Li, Runcong Zhao, Yongxin Yang, Yulan He, and Lin Gui. Overprompt: Enhancing chatgpt through efficient in-context learning, 2023.
- [18] Aayush Mittal. Zephyr-7b: Huggingface’s hyper-optimized llm built on top of mistral 7b, Nov 2023. URL <https://www.unite.ai/zephyr-7b-huggingfaces-hyper-optimized-llm-built-on-top-of-mistral-7b/>.
- [19] Shima Rahimi Moghaddam and Christopher J. Honey. Boosting theory-of-mind performance in large language models via prompting, 2023.
- [20] Aida Nematzadeh, Kaylee Burns, Erin Grant, Alison Gopnik, and Thomas L. Griffiths. Evaluating theory of mind in question answering, 2018.
- [21] Cathleen O’Grady, Christian Kliesch, Kenny Smith, and Thomas C. Scott-Phillips. The ease and extent of recursive mindreading, across implicit and explicit tasks. *Evolution and Human Behavior*, 36(4): 313–322, 2015. ISSN 1090-5138. doi: <https://doi.org/10.1016/j.evolhumbehav.2015.01.004>. URL <https://www.sciencedirect.com/science/article/pii/S1090513815000148>.
- [22] François Quesque and Yves Rossetti. What do theory-of-mind tasks actually measure? theory and practice. *Perspectives on Psychological Science*, 15: 384 – 396, 2020. URL <https://api.semanticscholar.org/CorpusID:211193800>.
- [23] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

- [24] Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning, 2019.
- [25] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL <https://aclanthology.org/D19-1454>.
- [26] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019.
- [27] Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large lms, 2023.
- [28] Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr, Yejin Choi, and Yulia Tsvetkov. Minding language models’ (lack of) theory of mind: A plug-and-play multi-character belief tracker, 2023.
- [29] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.
- [30] Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks, 2023.
- [31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.
- [32] Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. Open-tom: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models, 2024.

- [33] Lingjun Zhao, Khanh Nguyen, and Hal Daumé III au2. Define, evaluate, and improve task-oriented cognitive capabilities for instruction generation models, 2023.