Universiteit Leiden
The Netherlands

Bachelor Computer Science & Datascience and Artificial Intelligence

Evaluating the impact of Prompting Techniques on the logical consistency of LLMs

S.M. Frentz

First supervisor:
Dr. M.J. van Duijn
Second supervisor:
L.C. Froma MSc

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

20/02/2025

Abstract

This study evaluates the impact of various prompting techniques on the logical consistency and reasoning capabilities of Large Language Models (LLMs). Specifically, this is done in a Natural Language Inference (NLI) setting. To test this, four LLMs were chosen based on varying architecture types and parameter sizes: GPT-Turbo, Claude-3.5-sonnet, LLaMA-2-70B and Mistral-7B-Instruct. These models were tested on the LogiQA 2.0 dataset to assess their performance in producing accurate conclusions and coherent and correct reasoning chains. Logical consistency is a critical benchmark for testing the reasoning capabilities of LLMs. The prompting techniques that were used are: Zero-Shot, Few-Shot, Chain-Of-Thought and Generated Knowledge. Zero-shot prompting was used as a baseline because of its neutral characteristics. Results indicate that while prompting techniques do indeed influence performance, their overall impact on logical consistency in this specific setting is negligible. Few-Shot and Generated Knowledge produced modest performance improvements for GPT-3, but other models exhibited minimal or negative effects. The quality of the reasoning chains varied, with Claude out performing the other models in coherence and correctness. These findings stress the limitations of contemporary LLMs in complex reasoning tasks and highlight the need to future improvement and refinement of these prompting techniques to improve logical consistency for older models. While this study evaluates the impact of prompting techniques on logical consitency in LLMs, the used dataset has a binary gold label structure. This lacking a neutral category. This may have influenced the results by conflating contradiction with unrelated statements. Additionally, extracting the prediction labels themselves proved to be a non trivial task duo to the stochastic nature of LLMs. These limitations introduce uncertainty into the reported accuracy scores and reasoning evaluations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Large Language Model and Natural Language Inference

Large Language Models (LLMs), such as GPT-3 and GPT-4, have contributed to a significant improvement of the quality of AI by using models with billions of parameters to generate human-like text [1]. These models are trained on vast datasets often containing: books, webpages, code repositories, news articles and other publicly available web content. Large Language Models thrive in performing various natural language tasks, including Natural Language Inference (NLI). NLI refers to the process of determining whether a given hypothesis is true, false, or undetermined based on a given premise [2]. Understanding the mechanisms behind LLMs and their capabilities in NLI is crucial to exploring their limits and potential enhancements.

1.2 The importance and difficulties of reasoning

1.2.1 Reasoning as a feature

LLMs have excelled at many use cases for a long time. Auto-completion, new text generation in the style of another and many more. A use case that has always been a problem is abstract reasoning as stated by Bubeck et. al[3]. Bubeck disects reasoning as "finding and applying a general pattern from few data". Drawing new conclusions from previously given datapoints in a natural language setting aligns with human cognitive abilities to asses a situation. Doing more research in this field could provide a better insight in the perceived human-like coginitive abilities LLMs display.

1.2.2 Difficulties

The widespread use of LLMs has given rise to many new and innovative use cases. These new use cases vary from giving personal advice to fact checking. The big problem with many ambiguous use cases for LLMs is that there is a large unsolved problem that occurs in these models: hallucinations[4]. This phenomenon occurs when an LLM produces a piece of text which the users interpret as factually correct which in reality is actually false and completely fabricated. Hallucinations occur because LLMs are stochastic systems, as a result of which a sentence may be produced which is the most probable sentence given

the models' training data. However, the model has no inherit intuition for truth or false so other systems have to be put in place to check this output. There have already been numerous measures to counteract these hallucinations[5]. These hallucinations make some use cases more problomatic than others. Creating new ideas or texts define the power of the language models, whereas producing facts could be problematic because of the stochastic nature of these models. Reasoning is another one of these problematic use cases. For reasoning you need context or world knowledge and a good logical basis how one event will or can follow the previous event. Recent models, like DeepSeekV3, have been known to greatly improve these reasoning tasks by distilling larger complex reasoning capabilities to another smaller model[6].

1.3 Prompting

Just as presenting information in a new way, rephrasing the information may help humans understand the given message better. Prompting is a way to guide an LLM to the desired problem case. Prompting involes altering the given input to the model to a specific way to shape the response. This enables the model to perform specific tasks like reasoning, summarization or classification without the need for additional fine-tuning to this specific task. Certain structures[7] in this guidance have been defined to follow a set structure. Some of these techniques mirror the way humans learn through reinforcement and example for instance Zero-shot prompting, where no additional information is given, Few-shot, where some examples are given of the desired output.

1.4 Research Objectives

While looking forward with newer and bigger contemporary models, it is also interesting to look back and see if we can replicate the same results with older models. In this research I attempt to investigate the effectiveness of prompting techniques on the reasoning skills in last gerenation LLMs in an NLI setting. The goal is to expose these models to a dataset of logic-based NLI problems and let the models generate a reasoning chain and conclusion. Comparing these prompting techniques and the models could give interesting insights to what can help specific models perform better. The performances of the models will be quantified in two ways: a reasoning chain score and a conclusion label accuracy score.

Chapter 2

Theoretical Background

2.1 Large Language Models (LLMs)

Large Language Models (LLMS) have revolutionized the world of Artificial Intelligence by giving machines the ability to understand, generate and interact with natural language in ways that were previously never possible. The pivotal moment in this development was the introduction of the transformer architecture, first proposed in the paper *Attention is All You Need* by Vaswasni et al (2017)[8]. Traditional recurrent and convolutional neural networks were replaced by models with this transformer architecture by utilising a mechanism called self-attention. This process enables the models to filter all relevant information out of the input sequence, regardless of their position within this input. This proposed architectural style greatly improved the scalability and training times of these models.
The attention mechanism allows models to assign a level of importance to words or tokens in the input sequence. Letting the model capture contextual relationships of the input provides way to a new level of nuance understanding that was not possible before. This is combined with positional encoding, enabling the model to process sequences in parallel rather than sequential. Together, these two concepts addressed and solved the biggest bottlenecks older architectures, such as RNNs and LSTMs, faced.
Modern LLMs are variations of this proposed transformer architecture. Often choosing between an encoder-only or decoder-only, which both combine into the traditional transformer. Next to that, parameter size and fine-tuning approaches can also differ greatly which results in a big variance in area of expertise and performance of these models. OpenAI's GPT-3.5-Turbo and the newest GPT-4 are examples of decoder-only architectures, optimized for generating coherent and relevant text. On the other hand there are encoder-only models like BERT which are more suited for classification and question answering.
These advancements have placed LLMs at a fundamental position in the contemporary field of AI, capturing the world with high performance results in natural language translation, summarzation and natural language inference.

2.2 Natural Language Inference and Logical Reasoning

Natural Language Inference (NLI) is a task in natural language processing that involves letting a machine figure out the logical relationship between a premise and a hypothesis. More specifically, NLI classifies the relationship as either: entailment(the hypothesis logically follows the premise), contradiction(the hypothesis contradicts the premise) or neutral (the hypothesis cannot be determined from the premise and no correlation can be found). In Table 2.1 below a few examples of NLI classification are provided. This task is a fundamental test for a models' ability to reason and make inferences.

Premise	**Hypothesis**	**Label**
A man is sitting at a desk writing a letter.	The man is sleeping.	Contradiction
A woman is reading a book in the library.	The woman is studying for an exam.	Neutral
A soccer game with multiple males playing.	Some men are playing a sport.	Entailment

Table 2.1: Examples of Premises, Hypotheses, and their Logical Labels

The development of the field of NLI is historically not limited to LLMs but take a broader trend in AI. Early methods utilised a system of rules, which were handcrafted symbolic representations of the given context. While being easy to use and deduce, these rule systems struggled with scalability and handling more complex contexts. When AI started modeling after the human brain with machine learning and neural networks, statistical models helped with the automatization of the NLI task. However, they still could not obtain a deep contextual understanding of the given information and were limited by the features they were trained on.
When deep learning was introduced, neural networks took a more dominant place in NLI research. BERT and GPT are examples of models that make use of pre-trained representations of language to not only capture a representation of the literal word but also of the concept the word represents. This enables these models to deduce a more nuanced relationship between a given premise and hypothesis. Following this development, a big surge in performance increases were measured, particularly when these models were fine-tuned on large predetermined datasets such as SNLI and MNLI (Williams et al., 2018)[9]. As Wiegreffe and Pinter discuss in their publication *Explaining Simple Natural Language Inference* [2], even seemingly straightforward NLI tasks often require complex liguistic hints and logical structures, making them a fundamental benchmark for reasoning.

2.3 Prompting Techniques

The way information is presented to an LLM can make a big difference as to what the output entails and if the desired result is returned[10]. Together with the rise and popularity of Large Language Models has Prompt engineering also made it's way up and has been improved drastically. Prompting techniques, the practice of structuring a models' input to guide it's behaviour to a desired output, have been standardized and refined over the last few years. Prompting techniques, the method of changing your input to a certain structure, are central in this problem. More so-called prompting techniques have come up the more developments were made with LLMs. Research shows that different prompting strategies can significantly positively affect the accuracy and coherence of model outputs [7]. Examples of techniques are: zero-shot prompts, where the model generates responses without any additional context, few-shot and Chain-Of-thought prompts, which provide examples or structured reasoning patterns to enhance the models performance. Prompting techniques not only enhance a models' output but also serve as a bridge between initial intent and the interpretation a machine has of this intent. This draws a great parallel to cognitive strategies like framing and priming. The choice of prompting technique can impact the models understanding of context, logical reasoning, and ability to infer, which are all crucial skills in NLI tasks [2].

Chapter 3

Methodology

3.1 Experimental Design

The setup of this study is an NLI environment based on logic given a premise and a hypothesis structure. The basis on logic integrated in the used dataset is further expanded on in section 3.3.1. This study aimed to get a clear and wide comparison across multiple models(3.2) and prompting techniques(3.3.2). This was done to draw focus from one specific model to many different training and fine tuning sets. Prompting this environment creates a results dataset with outputs from different models and the predicted label given in the output text. In this study, Zero-shot was chosen as a baseline comparison because it provides a clear and standardized reference point for all models. Unlike other technical systems where a true "zero-level" may be defined, the differences in training data, architecture and design goals make a true "zero-level" impractical and difficult to find. A Zero-shot setting provides and unaffected playing field for reasoning. This serves as a neutral starting point for evalualtion. Taking this approach give all models a fair comparison, regardless of underlying differences. Lastly, the quality of reasoning is measured. By prompting a state-of-the-art model from openAi, gpt-4o, with the generated reasoning it attaches a reasoning score to each individual reasoning chain. This is done to quantify how well each model performed on reasoning given each specific prompting technique.

3.2 Models Used

For this study I selected four models to compare them based on their performance in a logical rule-based NLI task. The models selected are Claude-3-5-Sonnet-20241022, GPT-3.5-Turbo, Llama-2-70B-Chat, and Mistral-7B-Instruct. They were selected to represent a diversity in parameter size, architecture type, and training approach. This was done to ensure I could evaluate how different models handle a logic-based premise-hypothesis type task.
Claude-3-5-Sonnet-20241022, Llama-2-70B-Chat, and GPT-3.5-Turbo are well known and widely-used models, mostly chosen for their all-around performance on NLP tasks. Finally, I also added Mistral-7B-Instruct, a significantly smaller model, parameter-wise, to understand how lightweight models handle these tasks as well. This Mistral model was specifically fine-tuned to follow specific instructions. This would make a fair comparison to see the trade-off of fine-tuning on a specific task and a significant decrease in parameter

count. The combination of these models allows for a good evaluation, including both smaller and larger models, instruction-following models, and general-purpose models, enabling me to draw concrete conclusions about the capabilities of current language models in logical reasoning tasks.

3.2.1 Claude-3.5-sonnet

Introduced in June of 2024[11], Claude-3.5-sonnet is part of the Claude-3.5 family of models together with Claude-3.5-haiku and Claude-3.5-opus. The parameter count of this specific model remains undisclosed but Anthropic's addendum specifiec that it has graduate-level reasoning (GPQA), undergraduate-level knowledge (MMLU), and coding proficiency (HumanEval). Anthropic has greatly invested in the world of Large Language Models with it's development of the entire line of Claude models. Anthropic has pitched the Claude-3.5 family as a robust and reliable solution for tasks requiring high-level reasoning, knowledge retrieval and structured problem-solving.

3.2.2 GPT-3.5-Turbo

The GPT-3.5 family of models build further upon the groundbreaking GPT-3 model line. GPT-3.5-Turbo, built by OpenAI, takes the strengths of its predecessors a step further with an undisclosed but updated and expanded parameter count and big improvements in efficiency and fine-tuning options. This specific model was designed to be a balanced trade off between performance and cost. One of the biggest advancements of GPT-3.5-Turbo, is its improved contextual understanding. Longer input sequences allow the model to take in longer and more complex conversations.

3.2.3 LlaMA-2-70B-Chat

In July of 2023 Meta introduced the Llama-2 line of models[12]. Multiple variants of base and fine-tuned models were released to the public for ease of use and specific use cases. Llama-2-70B-Chat is the largest of the line of models released with 70 billion parameters fine tuned on chat interactions. The fine-tuning process involved training the model on diverse datasets including conversational datasets. This was done to improve the models' ability to produce a human-like dialogue. This way, Meta made LlaMA a perfect fit for virtual assistants, interactive tools or customer support applications.

3.2.4 Mistral-7B-Instruct

Mistral-7B[13] was introduced as a lightweight, low parameter alternative model for bigger models such as LlaMA-2-13B and Llama-2-34B. In the introduction blogpost[14], Mistral-7B-Instruct was also introduced. Being fine-tuned on instruction datasets publicly available on Huggingface. Mistral accompanied the base model with this fine-tuned model to show that their base model can easily be fine-tuned. Mistral-7B-Instruct is a great example of the growing amount of smaller light-weight and easy to run models. These smaller models make a perfect tool for developers to deploy them in an environment with limited resources. This model is chosen in this study to better portray a range of performances and parameter counts.

3.3 Dataset Used and Prompt Design

3.3.1 Dataset: LogiQA 2.0

For this study a dataset was needed based around an NLI taskset and a logical problem structure. High levels of reasoning skills centered around world knowledge and logical implications were the two main characteristics this dataset needed. LogiQA 2.0 [15] fits all of these criteria. It is a large-scale logical reasoning reading comprehension dataset adapted from the Chinese Civil Service Examination. This dataset is based around Chinese examination questions and has been translated to English to result in an NLI dataset of 14752 instances. All of these instances are annotated with a gold label. These labels are only in either the form *entailed* or *not-entailed*, so the use of contradiction and neutral are combined. This dataset was chosen due to its high level of reasoning needed and instance density. In 2023 this dataset was also used as a benchmark in the study *Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4* [16]. Where it was used to asses the performance of ChatGPT and GPT-4 without the use of prompting techniques as a second element.

Limitations

LogiQA 2.0 uses a binary gold label system. This is a major limitation compared to the traditional NLI task. Seen as the neutral and the contradiction conclusions are merged into one label: not-entailed. Some examples that should be contradiction may have been misclassified as something else. This may confuse a model that is generally known to a NLL problem setting. This could lead to misleading accuracy metrics as models may have labeled cases that should have been contradiction to something else for the wrong reasons. For example if the LLM responded with: *"The reasoning supports the hypothesis but is not completely definitive"* or *"in some cases"* or *"Not neccesarily entailed"*, the actual conclusion given is very ambiguous and open to interpretation. Furthermore, this affects the predicted label extraction as in a case of *"Not neccesarily entailed"*. In this particular case it is difficult to systematically extract the right sentiment from the models' response.

Datapoint example

Shown below is a random example from the dataset. This particular example is about one's profession given the proportion of income one receives from it. In the example, the premise and hypothesis are marked as such, while the relationship is marked as "gold_label".

```
{
"idx": 218,
"premise": "Many people who call themselves teachers are not
    actually teachers because teaching is not their main
    source of income.",
"hypothesis": "A person cannot be called a writer unless the
    writing is his main source of income.The same is true
    for teachers.",
"gold_label": "entailed"
}
```

3.3.2 Prompting Techniques

In this study I have used four prompting techniques to accurately test and compare the performance of different models: Zero-shot(ZS), Few-shot(FS), Chain-Of-Though(CoT) and Generated Knowledge(GK).

Few-shot Prompting

In a Few-shot setting, some examples of context related structures are given to the LLM. This helps the LLM learn during runtime and leverages in-context learning. Brown also found that this improves performance[17]. Below is a basic example of a few-shot setting on a logical premise-hypothesis task.

```
Example 1:
Premise: All humans are mortal. Socrates is human.
Hypothesis: Socrates is mortal.
Answer: Entailed

Example 2:
Premise: Some birds can fly. Penguins are birds.
Hypothesis: Penguins can fly.
Answer: Not Entailed

Now, answer the following:
Premise: All cats are mammals. Some mammals are nocturnal.
Hypothesis: Some cats are nocturnal.
Answer:
```

My prompt used for the few-shot setting was made with predetermined examples and looks much alike the Zero-Shot setting. The difference between these two prompts is found the examples that are given at the beginning of the prompt:

```
You are tasked with analyzing the relationship between
a premise and a hypothesis.
Your output must strictly follow this format:
Reasoning: <Provide reasoning here>
Conclusion: <entailed OR not-entailed>
Here are some examples:
Example 1:
Premise: The sky is blue.
Hypothesis: The sky is not blue.
Reasoning: The premise states that the sky is blue, but
the hypothesis contradicts this by saying it is not blue.
These statements cannot both be true at the same time.
Conclusion: not-entailed
Example 2:
Premise: A dog is barking.
Hypothesis: There is a dog barking.
Reasoning: The hypothesis directly repeats information
```

```
from the premise, so it logically follows.
Conclusion: entailed
Example 3:
Premise: A person is playing soccer.
Hypothesis: The person is doing a physical activity.
Reasoning: Playing soccer is a form of physical activity,
so the hypothesis logically follows from the premise.
Conclusion: entailed
-----
Now analyze the following:
Premise: {premise}.
Hypothesis: {hypothesis}.
Reasoning: <Your reasoning here>
Conclusion: <entailed OR not-entailed>
```

Zero-shot Prompting

In a zero-shot setting, the model is only provided with the problem statement, in this case the premise and hypothesis, without any additional context or examples. This technique tests the model's direct reasoning skills. Zero-shot prompting has been formally introduced in the groundbreaking paper *Language models are few-shot learners* by Brown et. al[17]. This paper shows that an LLM, not fine-tuned on a specific task, can yield acceptable results in different environments given the structure of the prompt. An example of Zero-shot prompting is shown below.

```
Premise: All cats are mammals. Some mammals are nocturnal.
Hypothesis: Some cats are nocturnal.
Is the hypothesis logically entailed by the premise? Answer:
```

In this study I used this prompt to create a Zero-shot environment for the given premise and hypothesis:

```
You are tasked with analyzing the relationship between
a premise and a hypothesis.
Your output must strictly follow this format:
Reasoning: <Provide reasoning here>
Conclusion: <entailed OR not-entailed>
----
Analyze the following:
Premise: {premise}.
Hypothesis: {hypothesis}.
Reasoning: <Your reasoning here>
Conclusion: <entailed OR not-entailed>
```

This prompt gave the model a simple instruction and ensured a specific output to make analyzing the data later easier.

Chain-Of-Thought (CoT)

Chain-Of-Thought prompting encourages the model to break down its answering process into reasoning steps. This technique improves interpretability and often improves accuracy. This was shown in the paper *Chain-Of-Thought Prompting Elicits Reasoning in Large Language Models* by Wei et. al [18]. In this paper Wei gives a clear structure how Chain-Of-Thought prompting can be achieved and why it is an improvement from zero-shot or few-shot settings. A flowchart illustrating the reasoning process for Chain-Of-Thought is shown in Figure 3.1. Breaking down the problem in clear and concise reasoning steps may help the model understand the problem setting better. This helps the model to take all factors into account while generating an output.

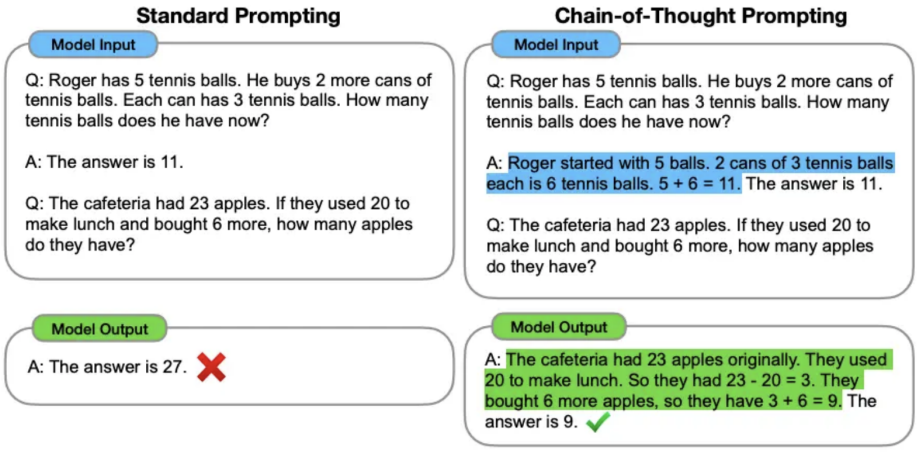

Figure 3.1: Example of Chain-Of-Thought Prompting. Image source: Wei et al.[18]

To ensure the generation of clear and concise reasoning steps, I used the prompt below.

```
You are tasked with analyzing the relationship between a
premise and a hypothesis.
Your output must strictly follow this format:
Reasoning: <Provide clear step-by-step reasoning here>
Conclusion: <entailed OR not-entailed>

Now, analyze the following thinking step by step:
Premise: {premise}.
Hypothesis: {hypothesis}.
Reasoning: <Provide your reasoning steps here, explaining
why the premise and hypothesis are related or not.
Be explicit and logical in your thought process.>
Conclusion: <After reasoning through the steps, conclude
whether the relationship is entailed or not-entailed in a single word.>
```

Generated Knowledge (GK)

The last prompting technique that was used in this study is Generated Knowledge (GK). Generated Knowledge, as the name states, incorporates extra knowledge about the context of the hypothesis and premise when prompting the final question. This information is used to enrich the context used by the model and improve the accuracy by including real world knowledge. The flowchart 3.2 below described the process of creating a Generated Knowledge setting as shown in the paper *Generated Knowledge Prompting for Commonsense Reasoning* by Liu et. al [19]. This flowchart breaks down how there are two clear steps to GK-prompting:

1. Ask the model to generate knowledge about the problem setting. The problem setting functions as the hypothesis and premise in this case.

2. Prompt the model again with the generated knowledge and the final question.

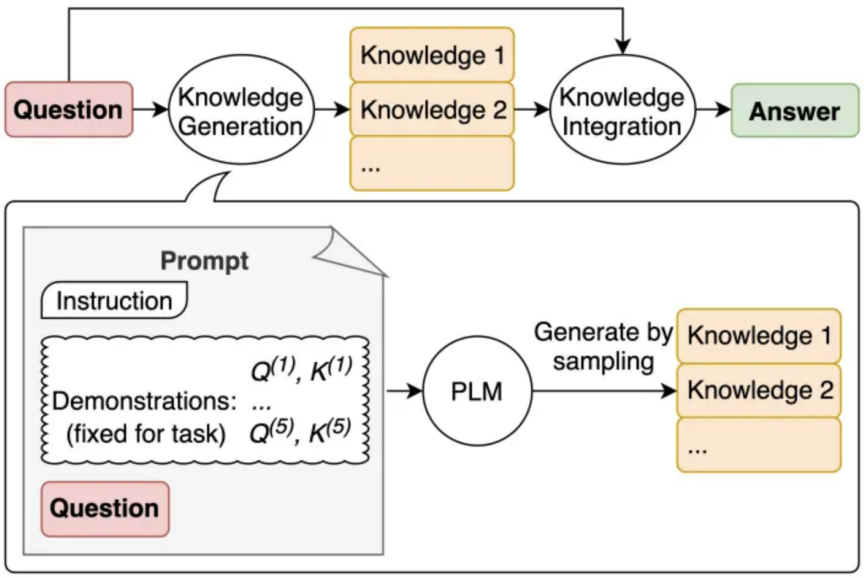

Figure 3.2: Flowchart of Generated Knowledge Prompting. Image source: Liu et al.[19]

To follow these steps coherently I used two prompts; one to generate the knowledge and one the generate the NLI reasoning and label.

```
Input: Greece is larger than mexico.
Knowledge: Greece is approximately 131,957 sq km, while
Mexico is approximately 1,964,375 sq km, making Mexico
1,389% larger than Greece.
Input: Glasses always fog up.
Knowledge: Condensation occurs on eyeglass lenses when water
vapor from your sweat,
breath, and ambient humidity lands on a cold surface, cools,
and then changes into tiny drops of liquid,
forming a film that you see as fog. Your lenses will be
relatively cool compared to your breath, especially when the
outside air is cold.
Input: premise:'{premise}'.
Hypothesis: {hypothesis}
knowledge:
```

The result of this prompt is real world knowledge about the specific hypothesis and premise. This knowledge was then used in the following prompt to create the desired output:

```
You are tasked with determining whether the relationship between
a premise and a hypothesis is entailed or not-entailed.
Here is the knowledge generated about the premise and hypothesis:
{knowledge}
Using this knowledge, analyze the logical connection between the
premise and hypothesis.
Identify whether the hypothesis logically follows from the premise.
```

```
Provide clear reasoning for your analysis and a definitive conclusion.
Premise: {premise}
Hypothesis: {hypothesis}
Reasoning: <Provide reasoning here>
Conclusion: <entailed OR not-entailed>
```

3.3.3 Prompt Design Considerations

The design of the prompts was chosen to align with the logical structure of the task at hand, making sure the models could effectively process the premise and hypothesis and give each datapoint and equal chance. The following considerations were of great importance when designing the prompts:

- **Clarity in premise-hypothesis pairs**: It was fundamental that for each prompting technique, the premise and hypothesis were both clearly defined and annotated to avoid any ambiguity. So the model clearly sees what the hypothesis is and what the premise is. This clarity avoids any variance in results that may arise because of longer or shorter premise-hypothesis pairs or difficult language used.
- **Balancing brevity and informativeness**: For the Zero-Shot and Few-Shot prompts, it was a high priority to create a balance between brevity and informativeness. The prompts needed to be concise enough to avoid overwhelming the model and flooding its context limit, while still providing it with sufficient information to comply to the technique and guide its reasoning.
- **Iteratively refining Chain-Of-Thought and Generated Knowledge prompts**: The prompts for Chain-Of-Thought and Generated Knowledge were constructed iteratively to make sure logical consistency and coherence were present in the reasoning chains.

These considerations aimed to minimize bias and maximize the interpretability of the model's reasoning process. All of the prompts were designed by the design given in the papers where they were introduced and mentioned in Paragraph 3.3.2

3.3.4 Analysis Methods

To analyze the given results of the different prompting techniques multiple evaluation methods were used. Accuracy evaluations with statistical significance are paired with a Reasoning score. The combinations of these two methods provide a independent assesment of the results.

Accuracy Evaluation

The accuracy is measured as the proportion of correctly predicted labels by each model across the dataset. The generated conclusion label given the premise-hypothesis pairs, is compared to the gold labels in the LogiQA 2.0 dataset.
To establish an accurate baseline for this measure, *Zero-Shot* will be used as a reference point. The performance of other techniques will be compared to this performance.

Statistical Significance

To say something accurate about the difference in results given by the prompting techniques, a statistical analysis will be done. Especially a *paired t-test* will be conducted. The test will comapare the accuracy of the prompting techniques to the Zero-Shot baseline.
When the T-test produces a significance level of $p < 0.05$ will be categorised as significant and statistically meaningfull. If the compared p-value will be higher than this, the difference is statistically insignificant. This will give an objective assesment of the measured results.

Accuracy vs. Prompt Length and Verbosity Bias

Given that different prompting techniques may deliver different length prompts, an analysis on this relationship will also be conducted. This analysis tracks the accuracy(binned into predefined ranges) across the prompt length(measured in character count) and will show if there is a direct correlation between the two.
A LOWESS (*Localy Weighted Scatterplot Smoothing*, as introduced by Cleveland(1979)[20]) regression will be applied to visualize the relationship. In addition to that, a standard deviation will be computed within each bin to assess the variance in the given bin.
Given that verbosity bias—where models tend to favor responses based on length rather than content—could influence performance, the following measures are taken:

- The distribution of prompt lengths across different prompting techniques will be analyzed.
- The accuracy of different models will be examined across length bins to identify a relationship.
- Bins with fewer than 10 samples will be marked as *Low N* to indicate that the bin is underrepresented.

This analysis will help to show that the changes in performance are due to the used prompting techniques rather than the models' preferences for longer prompts.

Reasoning Quality Score

A second measure to evaluate the responses of the models is a Reasoning Quality Score. This score will assess the coherence and correctness of the reasoning and explanation given by the model in response to the prompt. This will be done to see if the explanation a model can give can differ from the actual conclusion label.
The reasoning quality score will be assigned by the LLM *GPT-4o-Turbo*. The score ranges form 0 to 1 and is an indication of the coherence and correctness of the given explanation. For each model and prompting technique, the average reasoning score is calculated along with a high-score precentage. This is a percentage of how many instances had a reasoning quality score of 0.8 or higher.

Chapter 4

Results

This section presents the results of evaluating four language models—Claude-3.5, GPT-3.5-Turbo, Llama-2-70B, and Mistral-7B—on logical reasoning tasks using four prompting techniques: Zero-Shot, Few-Shot, Chain-Of-Thought, and Generated Knowledge. The goal is to assess the impact of these techniques on model accuracy and reasoning coherence.

4.1 Performance Analysis

The first part of the analysis consists of an accuracy reading of the NLI dataset with the given prompting techniques. For every prompting technique I calculated an accuracy score over all 2000 entries in the dataset. As Zero-shot was used as a base case, I compared the other prompting techniques' scores to the accuracy of Zero-Shot. The accuracy scores observed in the Table4.2 below must be interpreted in the context of the dataset's two-label limitations. Since contradiction and neutral are merged, models may have struggled to distinguish between logically contradictory statements and statements that were just unrelated. This could have deflated or inflated the resulting accuracy score of the associated models' performance.

Model	Zero-Shot	Few-Shot	Chain-Of-Thought	Generated Knowledge
GPT	0.4965	0.5325 (+7.25%)	0.484 (-2.54%)	0.533 (+7.35%)
Claude	0.5600	0.5595 (-0.09%)	0.5765 (+2.91%)	0.5475 (-2.24%)
LLaMA	0.5535	0.5515 (-0.37%)	0.543 (-1.91%)	0.539 (-2.56%)
Mistral	0.5245	0.5115 (-2.48%)	0.5205 (-0.77%)	0.5075 (-3.24%)

Table 4.1: Average accuracy scores for each model and technique with percentage increase or decrease compared to zero-shot accuracy

4.1.1 Statistical analysis

To check if the differences are statistically significant, a T-test on the difference in performance compared to the Zero-Shot performance was performed. The average accuracy scores computed for each technique and compared to the Zero-shot baseline. The p-values of each test determined if the difference was statistically significant. A threshold of $p < 0.05$ was used to determine this. The results of these tests are shown in Table 4.2. As shown in the column on the right, only the GPT-3.5-Turbo model combined with the Few-Shot and

Generated Knowledge had a significant increase in performance. All of the other increases or decreases were not statistically significant.

Table 4.2: Accuracy Comparison Across Models and Techniques compared to Zero-shot performance

Model	**Technique**	**Accuracy (Zero Shot/Technique)**	**p-value**	**Significant**
GPT-3	FS	0.4965 / 0.5325	0.0227	**Yes**
GPT-3	CoT	0.4965 / 0.484	0.4292	No
GPT-3	GK	0.4965 / 0.533	0.0209	**Yes**
Claude	FS	0.5600 / 0.5595	0.9746	No
Claude	CoT	0.5600 / 0.5765	0.2923	No
Claude	GK	0.5600 / 0.5475	0.4266	No
LLaMA	FS	0.5535 / 0.5515	0.8988	No
LLaMA	CoT	0.5535 / 0.543	0.5048	No
LLaMA	GK	0.5535 / 0.539	0.3572	No
Mistral	FS	0.5245 / 0.5115	0.4108	No
Mistral	CoT	0.5245 / 0.5205	0.8001	No
Mistral	GK	0.5245 / 0.5075	0.2822	No

4.2 Analysis of Accuracy Across Prompt Lengths

Figure 4.1 shows the relationship between the length of the prompts (binned into character ranges) and the mean accuracy for all four prompting techniques: Zero-Shot, Few-Shot, Chain-Of-Thought and Generated Knowledge. On the primary y-axis of each subplot, the mean accuracy is shown across bins of the length of the prompts, with error bars representing standard deviation. The secondary y-axis, represented by gray bars, represents the sample size for each bin. Using LOWESS, smoothed trend lines provide a summary of the accuracy trends over the length of the prompt. When a point on one of the subplots is annotated by *Low N*, the particular bin was underrepresented in the resulting dataset because of insufficient data. A sample size of 10 or lower was chosen as the cutoff line.
Across all prompting techniques, there is a high variability in accuracy trends, with large standard deviations within each bin. For Zero-Shot and Chain-Of-Thought, accuracy stays relatively flat. This suggests the length of the prompt has a minimal impact of verbosity. On the other side, Few-Shot and Generated-Knowledge show a slight upward trend, indicating that the length of the prompt may have a beneficial effect on the accuracy. Most of the bins from 1500 characters and upwards are underrepresented and don't have enough samples to accurately say something about the upward trends.
Overall, the results do not indicate clear evidence for verbosity bias. While Few-Shot and Generated Knowledge did show an improvement with a higher character count, the variability and low samples sizes make it unclear if the length of the prompt is a direct cause.

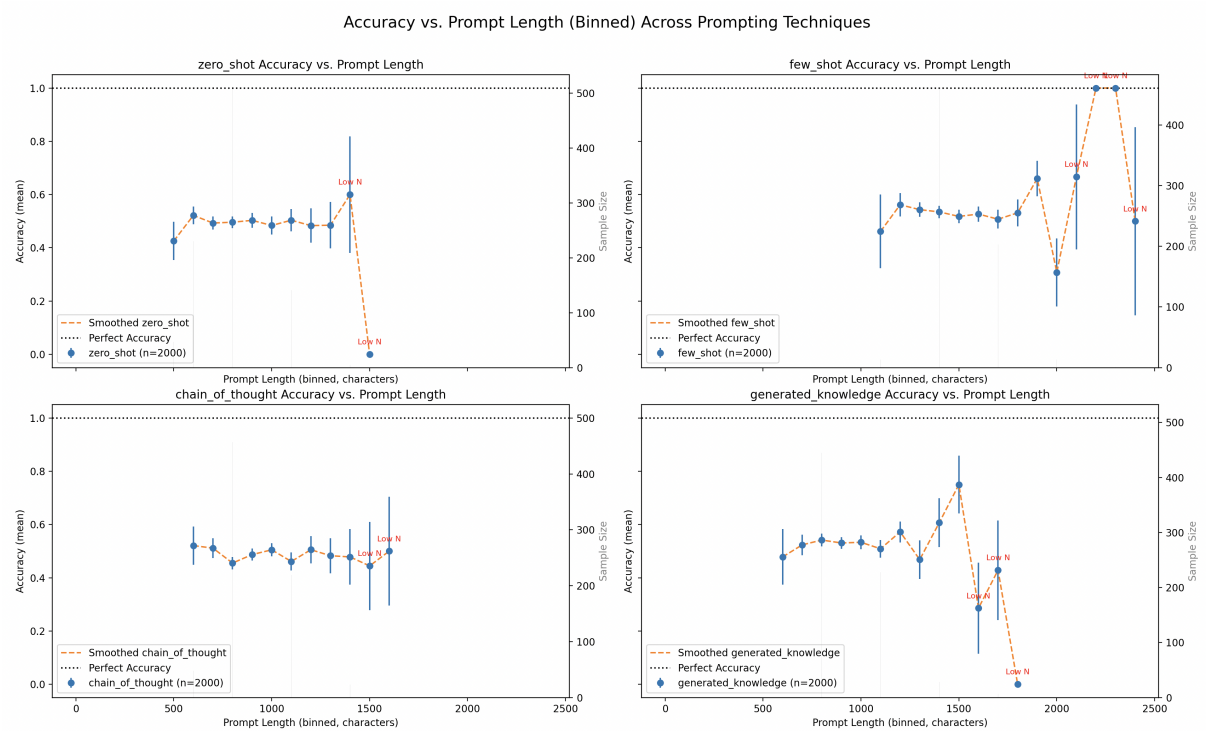

Figure 4.1: Accuracy vs. Prompt Length (binned) across prompting techniques. Each subplot represents a prompting technique, showing the mean accuracy with error bars for standard deviation, smoothed trends, and sample sizes. Low sample sizes are marked where applicable as *Low N*.

4.3 Reasoning Quality Evaluation

The last part of analysing how the models behaved is a reasoning quality score. This score is generated by GPT-4o and assesses how coherent and correct the reasoning chain is regardless of the outcome label. In Table 4.3 there is a detailed overview of the resulting data. This table presents the reasoning scores across the four used models(GPT, Claude, LLaMA, and Mistral) for the four prompting techniques used: Zero-Shot, Few-Shot, Chain-Of-Thought and Generated Knowledge. The left column is the average reasoning score for all reasoning chains that model has produced for a certain prompting technique, and the right column is the percentage of high performing reasoning chains in the sample. Overall, Claude outperforms the other models in both average reasoning scores and high-score percentage. Especially in the generated knowledge setting, having a near perfect(99.85%) high-score percentage. GPT-3.5-Turbo shows a stable performance for all prompting techniques, with a clear increase in the generated knowledge setting as well. LlaMA follows GPT-3-Turbo in its consistency, averaging just above GPT-3-Turbo. Mistral, however, consitently performs the lowest of all models. Especially in the generated knowledge setting, with only 15.85% higher scores than 0.8. While the reasoning score is a reflection of the coherence and correctness of the models' response, it is important to acknowledge that the limitation from a tri-label NLI setting to a binary NLI setting probably has influenced either the reasoning chain or the tone of the reasoning chain generated by these models. This may have influenced the reasoning scores and this should be taken into consideration when interpreting these results.

Model	Average Reasoning Score	High-Score Percentage (≥ 0.8)
ZS Prompting		
GPT3	0.75	60.55
CLAUDE	0.88	98.60
LLAMA	0.80	76.40
MISTRAL	0.71	45.55
FS Prompting		
GPT3	0.76	61.15
CLAUDE	0.87	96.75
LLAMA	0.76	62.35
MISTRAL	0.61	17.95
COT Prompting		
GPT3	0.76	65.95
CLAUDE	0.88	97.60
LLAMA	0.83	84.40
MISTRAL	0.66	34.15
GK Prompting		
GPT3	0.81	83.05
CLAUDE	0.90	99.85
LLAMA	0.71	70.75
MISTRAL	0.55	15.85

Table 4.3: Comparison of reasoning scores and high-score percentages (≥ 0.8) across the four prompting techniques.

Chapter 5

Discussion

5.1 Key Findings

5.1.1 LLMs stengths and weaknesses

The analysis of the resulting data revealed a few key characteristics that will be discussed in this chapter. The models displayed varying levels of accuracy across different reasoning contexts with different prompting techniques. Notably, Claude performed best in the Zero-Shot setting, indicating the model may have a better baseline reasoning performance without the addition of context.
However, only one of the models showed a significant improvement in performance after being provided with the additional prompting strategies and context information.
GPT-3-Turbo showed some improvement in the Few-Shot and Generated Knowledge settings, suggesting more examples and in-context knowledge could be reasons for a performance increase for this specific model. But there was also a decrease in performance in the Chain-Of-Thought setting, giving various results. This may suggest that for GPT-3-Turbo a more direct and clearly defined prompt design may be beneficial but more complex reasoning chains may be too impactful on performance and solely hurt the logical reasoning capabilities of this model.
LLaMA and Mistral both showed very consistent results on all prompting techniques in the study. Both models experienced a purely negative effect on the performance for all used prompting techniques. This may illustrate that the models are not well equipped for reasoning tasks in such a complex setting or in a multistep reasoning chain environment with the addition of these prompting techniques. However, all of the decreases were statistically insignificant so it is not clear evidence that it impacts their logical reasoning capabilities.

5.1.2 Impact of prompting techniques

The study results show that the impact of prompting techniques on a reasoning and logic based NLI setting are more limited than beneficial. While the different prompting techniques used did make an impact, the average accuracy impact was only 0.11%. Other than the outlier, being GPT-3-Turbo, most increases or decreases were modest and did not significantly impact the models reasoning capabilities.

Few-Shot

Few-Shot prompting had a notable performance improvement for the model GPT-3-Turbo, where the addition of a few examples increased the performance by 7.25%. While the other models only decreased in performance with the addition of examples. This indicates that giving examples can increase the models' performance but it is still difficult to get a consistent baseline in the reasoning NLI setting. The increase in performance is however an indication that giving GPT-3-Turbo examples may help with its logical reasoning skills and makes the model more consistent.

Chain-Of-Thought

While this prompting technique is meant to encourage the model to reason in a more structured and explicit way, this effect did not show in this study. In fact, most models including GPT-3-Turbo suffered from the added complexity. This may suggest that the added structure and thinking steps hindered the models ability to stay on track and accurately assess the problem in question. This highlights the potential trade-off between generating explicit reasoning and producing an accurate conclusion label.

Generated Knowledge

In contrast to giving examples of how the problem setting is syntactically supposed to be solved, Generated Knowledge aims to help a model semantically by giving additional context related world knowledge. This in itself is also a test in keeping this knowledge in active context and letting it help the reasoning and label generation. The results show that it has a comparable impact as the Few-Shot setting with the negative impact being greater. While GPT-3-Turbo benefits from the additional knowledge, Claude, LLaMA and Mistral had a bigger negative impact on the accuracy. What is interesting is that the average reasoning scores greatly increased in this setting for the models GPT-3-Turbo and Claude. This suggests that while the additional knowledge may interfere with a clear sight on the conclusion, it did help with a more coherent reasoning chain.

Overall

In summary, the data suggests that while prompting techniques do indeed influence a models' performance in some problem settings, their impact on this reasoning NLI dataset was in most cases neglegable. This is also confirmed by the statistical analysis in Paragraph 4.1.1. The trained ability for logical inference probably played a more central role in performing in this task rather than the additional context or examples. The results indicate that while prompting techniques may indeed have some beneficiary effects on a model's output, it does not help these last generation models overcome their reasoning difficulties and for most cases did not have a clear impact on their logical consistency.

5.1.3 Reasoning Score evaluation

The second measure of evaluation of the impact of prompting techniques on the reasoning capabilities of these models in an NLI setting is the Reasoning Score. These scores provide an independant evaluation of how the models' reasoning chain was coherent and correct regardless of the resulted outcome label. As was shown in the summary table 4.3, an average score was calculated and a percentage of high performing scores were displayed.

Few-Shot

The few shot prompting setting resulted in a mostly similar scoring result compared to Zero-Shot. This is in contrast to the label accuracy where Few-Shot made a definitive improvement for GPT-3-Turbo. These results show that although Few-Shot may have a beneficial impact on the accuracy, that does not necessarily mean the preceding reasoning chain is impacted at all.

Chain-Of-Thought

Chain-Of-Thought prompting also resulted in an increase in reasoning chain scores. The level of improvement varies by model. For GPT-3-Turbo, Chain-Of-Thought prompting made a slight improvement in average reasoning score and high-percentage score. This indicates that the way Chain-Of-Thought guides the model improved its reasoning consistency. Claude on the other hand shows a very consistent score for both Chain-Of-Thought and Zero shot prompting, suggesting that its reasoning capabilities are affected less by this technique. Overall, the results vary by model and do not give a clear indication of improvement. Often Chain-Of-Thought prompting enhances a model's reasoning capabilities though its effect is not universal and model dependant.

Generated Knowledge

Prompting in a Generated Knowledge setting resulted in the highest overall reasoning scores. Specifically Claude achieved an average score of 0.90 and a near 100 high(≥ 0.8) percentage of 99.85. This shows that Claude and GPT-3-Turbo benefited from the additional context knowledge to generate a coherent and correct reasoning chain.

5.2 Implications and Future Work

The results of this study highlight a part of Natural Language Processing where a lot of improvements can be made in the future. The flaws of contemporary LLMs are very clear in this logical reasoning setting. This results in some improvements to this study in the future and what other settings could be studied.

5.2.1 Model selection

As stated in the first chapter of this study, there have recently been many developments in the area of LLM reasoning. New training data and ever increasing parameter counts show that the reasoning gaps in LLMs can be solved that way. However, to make backwards compatable solutions we need to keep looking at older models. Not every machine is capable of running models with billions of parameters so this is still a valuable area of research.
While this study focussed on four low to mid end models, Claude-3.5, GPT-3.5, Llama-2, and Mistral-7B, it could give valuable insights to do the same study with different models and use these results to create a broader insight on models with these parameter counts.

5.2.2 Prompt engineering

The critical element in this study were the prompting techniques. Given the limited time and resources it was necessary to make some generalizations regarding the structuring of these prompts. For example the Few-Shot examples were manually curated for all context settings. In addition, giving more specific examples in the same syntactic manner would be an interesting future study.
Furthermore, Chain-Of-Thought prompting aims to encourage a model to give a structured reasoning chain. The results do not reflect this, therefore refining the structure and clarity may be beneficiary and improve the overall effectiveness of using this technique.

5.3 Reflection on Research limitations

This study has faced two major limitations that may have influenced the results coming forth from the research that was done. In this section these limitations will be discussed more deeply and reflected how they may have influenced the results and conclusion of this study.

5.3.1 Tri-label to binary label setting

The first limitation that impacted this study was the inherit flaw of the used dataset: LogiQA 2.0. This dataset, unlike a traditional dataset, uses a binary categorisation of the NLI problem task. Contemporary models that may be more familiar with a tri-label NLI task may get confused and respond differently when asked to categorise the relationship between a premise and a hypothesis when one of the options is missing. For now we can't say for sure that this has had a great influence but it should definetly be taken into consideration. Only future research with both a binary and a tri-label dataset could conclude whether or not this has really had a difinitive impact

5.3.2 Prediction label extraction

The second lmiitation that was faced in this study was the difficulties in extracting the predicted label from the resposne of the model. Because of the stochastic nature of LLMs, this task proved to be non-trivial. As seen in 3.3.2, a specific output structure was requested from the used models but because of this non deterministic nature, the responses weren't always in this format. This made extracting the one or two worded prediction label more difficult than predicted before. A series of regex functions and manual inspection have al yielded slighlty different results. This proved that the actual conclusionary labels and resulting accuracy scores could not be trusted fully. The differences were minor but could not to be proven fully correct. In a future rendition of this research, a big portion of the research time should be invested in the research of label extraction from a stochastic response.

Chapter 6

Conclusion

In this study, the impact of four prompting techniques on the logical consistency and reasoning capabilities of four enterprise level LLMs was evaluated: GPT, Claude, LLaMA and Mistral. The results that came out of the study showed that while prompting techniques did influence the performance of the models, the actual impact on logical consistency and reasoning coherence and correctness was modest at best, with only few significant improvements across all models. Claude consistently overperformed the other models, specifically in the Generated Knowledge setting, where it showed both high reasoning scores and a near perfect high-score percentage. GPT-3-Turbo demonstrated that the Few-Shot setting boosted its accuracy score but it did not exhibit corresponding improvements in reasoning scores, indicating that the presentation of examples may improve accuracy on deducing the right conclusion without necessarily making an improvement on the consistency of the reasoning given by the model. On the other hand LLaMA and Mistral showed very few signs of benefiting from the prompting techniques, suggesting they may struggle from the added complexity. These facts together show that the overall impact of prompting techniques was not clear on the logical consistency of the models.
This study underlines the limitations and choking points many LLMs face, while at the same time pointing out areas of future improvement. The results suggest that some prompting techniques may benefit from a different input structure, such as Generated knowledge, but further improvements on the techniques are definitely needed to accurately test their impact. Overall the results don't show a clear improvement on the logical consistency of these particular models given the prompting techniques as they were used. While these insights into the impact of prompting techniques on logical consistency in LLMs are provided by the results, the limitations should also be acknowledged. The dataset's two-label formate may have influenced these results by combining netraulity and contradiction, potentially affecting the validity of the accuracy scores. Additionally, the problems in extracting the actual prediction labels from the LLM-generated text introduced a layer of ambiguity. Future research should consider doing the same setup but with a trilabel dataset and refining the label extraction methods to ensure a more precise assesment of the reasoning capabilities of LLMs with the aid of prompting techniques.

Bibliography

[1] A. B. Yadav, "Generative ai in the era of transformers: Revolutionizing natural language processing with llms," *Journal of Image Processing and Intelligent Remote Sensing*, 2024.

[2] A.-L. K. et al, "Explaining simple natural language inference," *Proceedings of the 13th Linguistic Annotation Workshop*, 2019.

[3] S. Bubeck *et al.*, "Large language models are not strong abstract reasoners," *arXiv preprint arXiv:2305.19555*, 2023. [Online]. Available: `https://arxiv.org/abs/2305.19555`.

[4] R. Friel and A. Sanyal, "Chainpoll: A high efficacy method for llm hallucination detection," 2023. arXiv: `2310.18344 [cs.CL]`. [Online]. Available: `https://arxiv.org/abs/2310.18344`.

[5] K. Verspoor, "'fighting fire with fire' — using llms to combat llm hallucinations," eng, *Nature (London)*, vol. 630, no. 8017, pp. 569–570, 2024, ISSN: 0028-0836.

[6] DeepSeek-AI *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024. [Online]. Available: `https://arxiv.org/abs/2412.19437`.

[7] . I. G. D. P. M. C. J. W. S. A. A. H. P. R. Sander Schulhoff1 2 Michael Ilie1 Nishant Balepur1 Konstantine Kahadze1 Amanda Liu1 Chenglei Si4 Yinheng Li5 Aayush Gupta1 HyoJung Han1 Sevien Schulhoff1 Pranav Sandeep Dulepet1 Saurav Vidyadhara1 Dayeon Ki1 Sweta Agrawal12 Chau Pham13 Gerson Kroiz Feileen Li1 Hudson Tao1 Ashay Srivastava1 Hevander Da Costa1 Saloni Gupta1 Megan L. Rogers8 Inna Goncearenco9 Giuseppe Sarli9, "The prompt report: A systematic survey of prompting techniques," 2024.

[8] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," vol. 30, 2017.

[9] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1112–1122.

[10] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[11] Anthropic, *Claude 3.5 sonnet model card addendum*, Accessed: 2025-01-16, 2024. [Online]. Available: `https://paperswithcode.com/paper/claude-3-5-sonnet-model-card-addendum`.

[12] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023. [Online]. Available: `https://arxiv.org/abs/2307.09288`.

[13] M. AI, "Mistral 7b: A dense, efficient foundation model," *arXiv preprint arXiv:2310.06825*, 2023. [Online]. Available: https://arxiv.org/abs/2310.06825.

[14] M. AI, *Mistral-7b-instruct: A fine-tuned instruction-following model*, Accessed: 2025-01-16, 2023. [Online]. Available: https://mistral.ai/news/announcing-mistral-7b/.

[15] H. Liu, J. Liu, L. Cui, *et al.*, "Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2947–2962, 2023. DOI: 10.1109/TASLP.2023.3293046.

[16] J. Liu, K. Wang, Y. Qiao, M. Ding, Z. Liu, and M. Sun, "Evaluating the logical reasoning ability of chatgpt and gpt-4," *arXiv preprint*, vol. arXiv:2304.12808, 2023. [Online]. Available: https://arxiv.org/abs/2304.12808.

[17] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.14165.

[18] J. Wei, X. Wang, D. Schuurmans, *et al.*, *Chain-of-thought prompting elicits reasoning in large language models*, 2023. arXiv: 2201.11903 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2201.11903.

[19] J. Liu, A. Liu, X. Lu, *et al.*, "Generated knowledge prompting for commonsense reasoning," eng, 2021.

[20] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979. DOI: 10.2307/2286407.