FH University of Applied Sciences TECHNIKUM WIEN

# BACHELOR PAPER

Thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science in Engineering at the University of Applied Sciences Technikum Wien
Degree Program Computer Science Dual

# Comparing Chain-of-Thought, Chain-of-Verification and Self-Refine in solving Brazilian University Entrance Exams

By: Rafaela Rolim Santana
Student Number: 2210257114
Supervisor: Patrick Link, BSc.
Vienna, 02.04.2025

# Declaration of Authenticity

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz/ Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and according to the rules currently applicable at the UAS Technikum Wien and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.”

| | |
|---|---|
| Place, Date | Digital Signature |

# Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

Fusce aliquet pede non pede. Suspendisse dapibus lorem pellentesque magna. Integer nulla. Donec blandit feugiat ligula. Donec hendrerit, felis et imperdiet euismod, purus ipsum pretium metus, in lacinia nulla nisl eget sapien. Donec ut est in lectus consequat consequat.



**Keywords:** Keyword1, Keyword2, Keyword3, Keyword4, Keyword5

# Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.
Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.
Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

# Table of Contents

# 1 Introduction

In recent years, Large Language Models (LLMs) such as GPT-4 have demonstrated impressive capabilities across various tasks, including natural language understanding, code generation, and interactive dialogue systems. Despite these successes, a significant challenge remains: LLMs frequently produce plausible but inaccurate or misleading outputs, a phenomenon known as hallucination. Hallucinations can severely impact the applicability and trustworthiness of LLM-generated information, particularly in educational contexts. Students increasingly rely on LLM-generated answers for learning and assessment purposes; thus, inaccuracies can mislead their understanding, reinforce misconceptions, and ultimately hinder their academic progress.

To address these limitations, recent literature has explored innovative prompting strategies aimed at enhancing the reasoning and verification capabilities of LLMs. Notable methods include Chain-of-Thought (CoT), which encourages models to generate step-by-step reasoning sequences [1]; Chain-of-Verification (CoVe), where models explicitly verify intermediate reasoning steps [2]; and Self-Refine, which iteratively improves model outputs using self-generated feedback [3]. These methods show potential in systematically improving the reliability and accuracy of LLM-generated responses.

This thesis aims to conduct a comparative analysis of these three prompting methods, evaluating their effectiveness in improving answer accuracy in multiple-choice question-answering tasks. Specifically, the research seeks to answer the following questions:

- **RQ1**: How does the application of Chain-of-Thought, Chain-of-Verification and Self-Refine prompting techniques affect LLM response accuracy on Brazilian University Entrance Exams?
- **RQ2**: How does the effectiveness of Chain-of-Thought, Chain-of-Verification, and Self-Refine prompting techniques vary across different subject areas?

To achieve these goals, this thesis will adopt an experimental approach using standardized QA datasets. Experiments will compare quantitative measures such as general accuracy and accuracy by subject, and statistical analyses are performed to determine if significant performance differences exist between these techniques.

# 2 Methodology

## 2.1 Ideal Solution

An ideal solution for assessing prompting techniques would involve rigorously testing state-of-the-art methods using extensive, well-structured, and challenging question-answering datasets. There would be a reliable method to identify hallucinations or factual inaccuracies in the LLM responses. The optimal approach would identify the prompting technique that consistently yields the highest accuracy and lowest hallucination rate in LLM-generated answers, making it especially beneficial for educational applications where reliable knowledge dissemination is critical.

## 2.2 Requirements

Requirements were gathered through a comprehensive literature review focused on existing prompting techniques. The review included analyzing each method's documented strengths, weaknesses, and evaluation criteria previously employed in related studies. This helped establish the suitability of available datasets for meaningful comparisons, identify essential computational tools, and define the appropriate metrics required for a robust analysis.

## 2.3 Development Process

**Dataset Preparation**
Carefully selecting and curating a collection of standardized multiple-choice questions from the ENEM dataset. These questions include verified answers and are categorized by academic subject, facilitating detailed comparative analyses.

**Template Definition**
Clearly defining and creating consistent templates for each prompting method (Chain-of-Thought, Chain-of-Verification, Self-Refine). These templates guide the LLM response flow, ensuring method consistency across comparisons.

**Testing Phase**
Implementing each prompting method on the ENEM dataset. Multiple responses were systematically generated per question and method, and the resulting data was stored and organized for subsequent evaluation.

**Evaluation**

Conducting thorough comparative analyses of the generated responses to determine accuracy. This evaluation focused on overall method accuracy and subject-specific performance differences.

## 2.4 Tests

**General Accuracy**
Calculating the overall correctness percentage across all ENEM questions for each prompting technique.

**Accuracy by Subject**
Analyzing correctness rates within specific academic subjects to assess subject-dependent effectiveness.

**Consistency Score**
Determining the reliability of each method by calculating the percentage of times each prompting technique consistently produced the same answer across multiple attempts.

**Statistical Analysis**
Utilizing statistical tests (e.g., one-sided t-tests) to evaluate whether observed differences in accuracy among the prompting techniques were statistically significant.

## 2.5 Tools

All experiments were conducted within Jupyter Notebook environments, leveraging the OpenAI API. Data analysis and visualization employed Python libraries such as pandas, matplotlib, and scipy.

# 3 Solution

**How will I answer the research questions?**
I want to find out which prompting technique produces the most accurate results, or in other words produces less inaccuracies. For that I will use each technique to solve high-school level questions from the 2024 ENEM. I will make several test runs with each technique and then compare their results – what was the general accuracy, accuracy per subject. I will look for statistically relevant differences in performance of the three techniques.

## 3.1 Implement Templates for the Prompting Techniques

The implementation of prompting techniques involved defining templates for each of the three methods: Chain-of-Thought (CoT), Chain-of-Verification (CoVe), and Self-Refine. Each template was designed to guide the LLM to generate responses in specific structured formats to facilitate comparative analyses. These templates were programmed into the Jupyter Notebook and integrated with the OpenAI API, ensuring consistent application and interaction with the LLM.

### 3.1.1 3-shot Chain-of-Thought (CoT)

Chain-of-Thought (CoT) is a prompting technique designed to improve the reasoning abilities of large language models (LLMs) [1]. Instead of asking the model to answer a question directly, CoT prompting encourages it to think step-by-step, just like a human solving a complex problem. In a standard prompt, one might ask:

*Q: If there are 3 cars and each car has 4 tires, how many tires are there in total?*

And expect the model to respond:
*A: 12.*

With Chain-of-Thought prompting, the model is instead encouraged to reason like this:
*A: Each car has 4 tires. There are 3 cars. So, 3 × 4 = 12 tires in total.*

This method works particularly well for tasks that involve multi-step reasoning, such as mathematical problems, logic puzzles, or any task where intermediate steps are useful for reaching the final answer. The authors of the paper showed that CoT significantly improves the accuracy of models like GPT-3 on benchmark datasets like GSM8K for grade school math) [1]. They also found that CoT only works well with sufficiently large models [1] , like GPT-3. Smaller models, such as 13B or less, don't benefit as much because they struggle to produce coherent reasoning steps.

In this study, each CoT prompt included three examples of ENEM questions, each containing the reasoning steps leading to the correct answer. This reasoning was taken from verified high-school teachers' solutions, as described in this paper [4]. The 3-shot CoT approach prompts the LLM to explicitly outline each reasoning step when selecting the correct multiple-choice answer.

### 3.1.2 Chain-of-Verification (CoVe)

Chain-of-Verification (CoVe) is a prompting strategy designed to reduce hallucinations—confident-sounding but factually incorrect answers—in LLMs. Instead of just generating an answer and stopping there, the model verifies its own answer through a series of reasoning and checking steps. This method helps LLMs catch their own mistakes [2] by prompting them to reflect on the truthfulness and correctness of what they wrote. The process has four main parts:

**1.** Drafting
The model first generates an initial answer to the question.

**2.** Planning the Verifications
The model analyzes the draft and identifies which parts of it need to be checked. For example, if the answer includes facts, dates, or steps in a calculation, the model breaks those down into specific items for verification. This planning stage produces verification questions, such as:

- "Is Nix a moon of Mars?"
- "Did event X happen in year Y?"

**3.** Executing the Verifications
The model then answers each verification question separately. For each one, it uses a dedicated verification prompt and examines whether the original statement holds true through fact-checking, logic checks, or math reasoning.

**4.** Refinement
Finally, the model uses the verification results to revise the original answer if needed. If any verification step finds an issue, the answer is updated to fix it.

In this study, CoVe starts by generating an initial response using the 3-shot CoT template. The LLM is then prompted to generate two to four verification questions for the initial response. Each verification question is answered individually. Finally, the verification questions and answers are used to generate a final, refined answer.

### 3.1.3 Self-Refine

Self-Refine is a method that improves the quality of outputs from large language models by letting them revise their own responses through feedback and refinement [3]. Instead of generating a final answer in one go, Self-Refine works in feedback iterations. The same model plays three roles: it generates the initial response, gives feedback on it, and then improves it

based on that feedback. This process continues until the result is good enough or a stopping condition is met. The method has three main steps:

**1.** Initial Generation
The model first produces an answer to the input. This is the initial draft $y_0$.

**2.** Feedback
Next, the model reads its own output $y_0$ and writes a feedback message about it. This feedback should be actionable (it suggests specific changes) and specific (it points out exactly what to improve), for example:
*"This code uses a loop to add numbers, but it can be optimized using a formula."*

**3.** Refinement
The model then uses the feedback to revise the original answer. It tries to fix mistakes or improve clarity, style, or correctness. This step produces a new version of the answer ($y_1$).
The feedback and refinement process can repeat multiple times—each time building on the last version—until no more improvement is needed.

In this study, Self-Refine first generates an initial response using the 3-shot CoT template. The LLM is then prompted to analyse this answer, give it critical feedback, and generate a refined answer with reasoning chain taking this feedback into account. This process continues until the refined answer chooses the same MC alternative as the previous answer.

## 3.2 Load QA dataset

The dataset selected for this study consisted of multiple-choice questions from the 2024 ENEM exam, a standardized test widely recognized and used in Brazil. These questions offer a structured and standardized format, making them suitable for evaluating the accuracy of different prompting techniques across various academic subjects.

The dataset was provided in JSON format by the Maritaca AI team on Hugging Face [4], with each entry containing a question, the multiple-choice alternatives and the correct answer. The dataset was loaded into a Jupyter Notebook using Python's pandas library.

Initial preprocessing included parsing JSON data into a pandas DataFrame for easier manipulation, extracting relevant fields such as the question text, answer choices and correct answer, and setting subject labels to each question.

## 3.3 Run Tests

After implementing the prompting templates, the experimental tests were systematically conducted. Each prompting technique was individually applied to the entire QA dataset, ensuring comprehensive coverage across all subjects.

The testing process involved sending each multiple-choice question individually through the OpenAI API, formatted according to each respective prompting technique. Several test runs allowed for the collection of multiple responses per question to assess consistency. All generated responses were stored in csv files, containing corresponding metadata such as question identifiers, prompting method used, trace of feedback iterations, chosen answers and timestamp.

## 3.4 Clean Data

After running the tests and collecting responses, the datasets required further cleaning and preprocessing to facilitate the performance analysis. Data cleaning was an essential step to remove irrelevant information generated during the experimental runs.

## 3.5 Analyse Performance

The final step involved analyzing the cleaned datasets to evaluate the performance of each prompting technique. This phase focused on quantitatively assessing accuracy and reliability through statistical comparisons. Key steps in the performance analysis included:

**General Accuracy Assessment**:
Calculating the overall accuracy percentage of each prompting technique, determining how frequently each method produced correct answers across the entire dataset.

**Subject-specific Accuracy Analysis**:
Breaking down accuracy scores by subject area (Mathematics, Natural Sciences, Human Sciences and Languages) to identify potential strengths or weaknesses of prompting techniques across different academic domains.

**Consistency Evaluation**:
Evaluating how consistently each method provided identical answers upon repeated querying, thus reflecting the reliability and stability of the prompting methods.

**Statistical Significance Testing**:
Performing statistical tests to ascertain whether the differences in performance observed

between prompting techniques were statistically significant. Python's scipy library was employed to conduct these tests and calculate p-values, confirming whether observed differences could be considered meaningful or were merely due to random variation.
This rigorous analytical process ensured objective comparisons between Chain-of-Thought, Chain-of-Verification, and Self-Refine techniques, providing clear evidence of their relative effectiveness within educational contexts.

## 3.6 Guidelines

# 4 Discussion

## 4.1 Potentials

## 4.2 Limitations

## 4.3 Possible Improvements

# 5 Bibliography

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le und D. Zhou, „Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems 35,* 2022.

[2] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz und J. Weston, „Chain-of-verification reduces hallucination in large language models," arXiv preprint arXiv:2309.11495, 2023.

[3] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck und e. al, „SELF-REFINE: Iterative Refinement with Self-Feedback," *Advances in Neural Information Processing Systems 36,* 2023.

[4] D. Nunes, R. Primi, R. Pires, R. Lotufo und R. Nogueira, „Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams," arXiv2303.17003, 2023.

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le und D. Zhou, „Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems 35,* 2022.

[2] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz und J. Weston, „Chain-of-verification reduces hallucination in large language models," arXiv preprint arXiv:2309.11495, 2023.

[3] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck und e. al, „SELF-REFINE: Iterative Refinement with Self-Feedback," *Advances in Neural Information Processing Systems 36,* 2023.

[4] D. Nunes, R. Primi, R. Pires, R. Lotufo und R. Nogueira, „Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams," arXiv2303.17003, 2023.

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| WWW | World Wide Web |

# Documentation table of AI-based tools

| AI-based tools | Intended use | Prompt, source, page, paragraph... |
|---|---|---|
| **DeepL Translate** | Translation of an article in English | Source (XXX), Chapter X on page X-X |
| **ChatGPT (4.0)** | Grammar and spelling | "Please list issues with spelling and grammar in the following text: ..." Entire document |
| | | |

# A: Heading of Appendix A

# B: Heading of Appendix B