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 Prompting Techniques in 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.

Acknowledgements

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

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 This is the heading of the first chapter 5](#_Toc179533587)

[1.1 Heading level 2 5](#_Toc179533588)

[1.1.1 Heading level 3 5](#_Toc179533589)

[1.1.1.1 Heading level 4 5](#_Toc179533590)

[2 This is the heading of the second chapter 5](#_Toc179533591)

[2.1 Heading level 2 5](#_Toc179533592)

[2.1.1 Heading level 3 6](#_Toc179533593)

[2.1.1.1 Heading level 4 6](#_Toc179533594)

[Bibliography 7](#_Toc179533595)

[List of Figures 8](#_Toc179533596)

[List of Tables 9](#_Toc179533597)

[List of Abbreviations 10](#_Toc179533598)

[Documentation table of AI-based tools 11](#_Toc179533599)

[A: Heading of Appendix A 12](#_Toc179533600)

[B: Heading of Appendix B 13](#_Toc179533601)

# Introduction

## Motivation

Large Language Models (LLMs) are increasingly being used in educational contexts — from assisting with content generation to acting as intelligent tutoring systems. Their potential to provide immediate and personalized support makes them attractive tools for learning environments. However, for these models to be truly beneficial in educational settings, their outputs must be **reliable, accurate, and consistent**, especially when answering exam-style questions where factual correctness is crucial.

Students often rely on such models to help them **understand concepts, solve exercises, and prepare for exams**. Although LLMs can generate coherent text, their accuracy on exam-style questions varies widely. In this context, selecting an appropriate prompting technique may significantly influence the quality and correctness of the generated responses. Yet, there is currently limited guidance on which prompting methods are best suited for specific subjects or tasks.

Given the wide range of disciplines present in university entrance exams — such as Mathematics, Natural Sciences, Human Sciences, and Languages — it becomes essential to evaluate how different prompting strategies perform across domains. A systematic and subject-sensitive evaluation could help students and educators understand how to best leverage LLMs for educational use.

## ****Goals****

The goal of this thesis is to design and evaluate a framework for assessing the performance of different prompting techniques in solving standardized high-school level questions. More specifically, the objectives are:

* To investigate and compare three prompting techniques in terms of accuracy in solving Brazilian university entrance exams.
* To determine which subject areas benefit most or least from each technique.
* To provide guidelines for optimizing prompts to improve educational outcomes.

## Research Questions

 **RQ1**: How does the application of Chain-of-Thought [1], Chain-of-Verification [2] and Self-Refine [3] prompting techniques affect LLM response accuracy on Brazilian University Entrance Exams?

 **RQ2**: How does the effectiveness of Chain-of-Thought [1], Chain-of-Verification [2] and Self-Refine [3] prompting techniques vary across different subject areas?

## ****Expected Outcome****

The expected outcome of this work is a **framework** for evaluating and comparing prompting strategies using standardized questions and objective performance metrics.

**Moreover, this work aims to make recommendations** for students and educators on how to select and use prompting techniques effectively.

# Methodology

## Experimental Setup

This study uses a dataset composed of questions from the 2024 edition of the *Exame Nacional do Ensino Médio* (ENEM) [4], Brazil’s national university entrance exam. The dataset includes a total of 180 multiple-choice questions spanning four core subject areas: Languages, Human Sciences, Natural Sciences, and Mathematics. As the ENEM 2024 exam took place after the training cutoff date for the evaluated language model, it offers a robust benchmark with minimal risk of data leakage, thereby contributing to the integrity and validity of the experimental results.

All experiments are conducted using GPT-3.5 as the underlying language model. The implementation is carried out in a Jupyter Notebook environment, where the experimental pipeline is structured into distinct stages: data ingestion, prompt generation, response evaluation, and results analysis. This setup enables clear reproducibility and modular experimentation across different prompting strategies.

## Prompting Techniques

### 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.

### Chain-of-Verification (CoVe)

Chain-of-Verification (CoVe) [2] 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.

1. 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?”

1. 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.

1. 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.

### Self-Refine

SELF-REFINE is a method that improves the quality of outputs from large language models (LLMs) [3] by letting them revise their own responses through feedback and refinement.

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 using a normal prompt. This is the **initial draft** y₀.

1. **Feedback**

Next, the model **reads its own output** y₀ and writes a feedback message about it. This feedback should be a**ctionable** (it suggests specific changes) and s**pecific** (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.”*

1. **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 (y₁).

The feedback and refinement process can repeat multiple times—each time building on the last version—until no more improvement is needed.

## Implementation

Prompt templates were created for each of the prompting techniques, and the results were stored with each iteration for later analysis.

**Chain-of-Thought (CoT)**

In Chain-of-Thought few-shot prompting, each prompt contains:

* Examples of ENEM questions, including the reasoning steps and the final answer. These examples were taken from solutions of high school teachers to the ENEM questions [4].
* The question the model should answer along with the multiple-choice alternatives.
* Instructions to follow the examples and think step-by-step to solve the question.
* The desired response format.

**Chain-of-Verification (CoVe)**

In CoVe, an initial answer is generated using the CoT template. Then, verification questions for the initial answer are generated and executed individually. Finally, the initial answer is revised taking the verifications into account, providing the final answer.

**SELF-REFINE**

In Self-Refine, the first answer draft is generated using the CoT template. It then gives feedback to that answer and uses it to create a new, refined answer. The feedback loop continues until the refined answer contains the same alternative as the previous answer.

## Evaluation Metrics

**Accuracy**

* Answer accuracy: Percentage of correct answers compared to ground truth.
* Subject-wise accuracy: Performance breakdown by subject area.
* Comparative accuracy: Improvement over baseline (CoT).

**Consistency**

* Response consistency score: Percentage of identical answers across multiple runs
* Standard deviation of responses for each question

# Solution

# Discussion

Bibliography

|  |  |
| --- | --- |
| [1] | H. Balzert, Lehrbuch der Objektmodellierung - Analyse und Entwurf mit der UML 2, 2. Ausg., Elsevier GmbH, München 2005., München: Elsevier GmbH, 2005. |
| [2] | K. W. Wagner, Performance Excellence. Der Praxisleitfaden zum effektiven Prozessmanagement, München: Hanser Fachbuch, 2007. |

List of Figures

[Figure 1: Example of name and year printed on spine. 6](#_Toc330300567)

List of Tables

[Table 1: Schedule for “Applied Mathematics”. 6](#_Toc330300577)

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