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, 20.05.2025

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.”

Signed by:	Rafaela Rolim Sant'Ana
Date:	20.05.2025 15:47:25

A-TRUST

This document is digitally signed!

This document bearing a qualified electronic signature has the same legal standing as a handwritten document in accordance with Article 25 (2) of Regulation (EU) No. 910/2014 of 23 July 2014 (“eIDAS-VO”)

Validation: Information for checking the electronic signature can be found at: www.a-trust.at/pdf

Vienna, 20.05.2025

Place, Date

Digital Signature

Abstract

Large Language Models (LLMs) are widely used by students to solve academic tasks. While these models can be helpful, they often produce incorrect or misleading answers, a problem known as hallucination. This study explores whether simple prompting strategies can reduce hallucinations and improve answer accuracy without requiring fine-tuning or external tools.

Three methods were evaluated: Chain-of-Thought (CoT), which guides the model to reason step by step; Chain-of-Verification (CoVe), which adds a verification phase to check the initial answer; and Self-Refine, which lets the model review and revise its own responses through feedback loops. Each method was applied to 180 questions from the Brazilian university entrance exam ENEM 2024, covering four subject areas. The test was run 40 times per question to measure not only accuracy but also consistency across repeated runs.

The results showed that Chain-of-Thought achieved the highest overall accuracy and consistency, especially in subjects like Mathematics and Human Sciences. Self-Refine and CoVe showed more variability, and sometimes their refined answer performed worse than the initial answer.

Keywords: Large Language Models, Prompt Engineering, Chain-of-Thought, Chain-of-Verification, Self-Refine, Hallucination Mitigation, GPT-3.5 Turbo

Acknowledgements

I thank my parents, Márcio and Jane, and my sister, Fernanda, for believing in me and giving me strength to face this journey; and for always being present, despite the ocean between us. This would not have been possible without their support.

I thank my friends Thais, Guilherme and Victor for being great companions, bringing joy into my life and becoming my chosen family in Vienna. I thank my friends from Colegagi for their friendship that goes beyond time and distance. I thank Roger, who inspired me to study Computer Science and stood by me through challenging times.

I thank the lecturers, employees and colleagues from the FH Technikum Wien, for the support over these three years and for the great learnings. And finally, I thank my colleagues from A1 for the warm reception and for supporting my professional growth.

Table of Contents

1 Introduction

1.1 Motivation

The integration of Large Language Models (LLMs) into educational settings has accelerated rapidly since their release. Recent studies indicate that 70% of middle and high school students in the U.S. have used LLMs like ChatGPT for assignments across various subjects, including language arts and mathematics [1]. In higher education, over 90% of university students in the UK report using AI tools to support their academic work [2]. Educators are also embracing these technologies: a survey found that 56% of teachers incorporate AI tools into their teaching, though only a quarter have received formal training [3].

As a Computer Science student in the age of artificial intelligence, I have often relied on LLMs to support my study routine. These models act as accessible tutors, capable of answering questions, explaining concepts in simpler terms, and adapting to individual learning needs. Especially helpful is their non-judgmental nature, which encourages students to ask basic questions, revisit topics as many times as needed, and receive immediate feedback.

However, LLMs have a serious limitation: they sometimes produce outputs that are factually incorrect or misleading. These “hallucinations” may sound plausible, but can contain wrong or invented facts, being especially dangerous in educational contexts, where students might not yet have the background knowledge to detect such errors. This concern is widely recognized in the literature: if unnoticed, such hallucinations can lead to the reinforcement of false information and undermine learning outcomes [4] [5].

Despite this problem, LLMs continue to grow in popularity in education due to their accessibility and broad capabilities [6]. This raises an important question: how can we reduce hallucinations in a way that is simple, accessible, and does not require technical knowledge or system modifications?

To explore effective hallucination mitigation strategies, a literature review focused on state-of-the-art techniques was conducted. An overview of hallucination mitigation methods (see Figure 1) categorizes them into two main branches: *Prompt Engineering* and *Developing Models*. While model-based techniques like fine-tuning require infrastructure and model training expertise beyond the reach of typical users, prompt engineering methods stand out for their accessibility and low entry barrier. These methods can be applied directly using commercial APIs and chatbots, making them ideal for scenarios where students interact with LLMs as black-box systems.

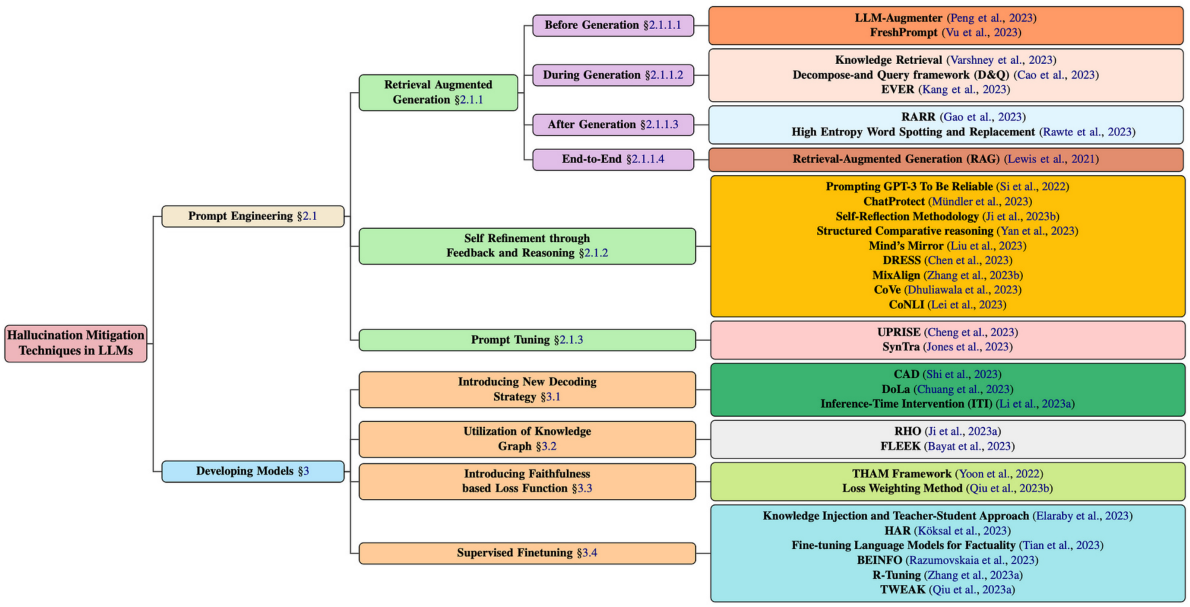

Figure 1: Imama, "Beyond Traditional Fine-tuning: Exploring Advanced Techniques to Mitigate LLM Hallucinations," Hugging Face Blog, Apr. 2024. [Online]. Available: https://huggingface.co/blog/Imama/pr. [Accessed: Apr. 10, 2025].

Within prompt engineering, three sub-categories are outlined: *Retrieval-Augmented Generation (RAG)*, *Prompt Tuning*, and *Self-Refinement through Feedback and Reasoning*. Based on feasibility, the first two were excluded: *RAG* relies on external knowledge bases or search modules [7], which a student may not have access to, and *Prompt Tuning* requires gradient-based optimization and training tokens [8], which is impractical for everyday users.

This narrowed the focus to self-refinement methods: a subset of prompt engineering that enhances LLM performance through structured reasoning and self-improvement. Among the techniques surveyed, three suitable candidates were identified, being both effective and feasible:

Chain-of-Thought (CoT)
Encourages models to produce intermediate reasoning steps, improving performance on complex multi-step problems [6].

Chain-of-Verification (CoVe)
Adds a verification phase that critically assesses and revises the model's initial answer, shown to reduce hallucinations [9].

SELF-REFINE
Implements an iterative loop of feedback and refinement, helping models self-correct reasoning and improve factual accuracy [10].

These techniques were selected because they:

1. Require no model fine-tuning or external tools,
2. Operate entirely through prompting, and

3. Can be reproduced with standard access to the OpenAI API.

The design of the experiment was also shaped by the evaluation needs of these techniques. To assess factual correctness, the ENEM 2024 university entrance exam [11] was selected. It contains 180 multiple-choice questions across four subject domains, along with the ground-truth answers. This allowed for straightforward, automated scoring without the need to judge the reasoning quality, ideal for isolating the effects of prompt design alone.

If prompt-based strategies can reliably reduce hallucinations in this setting, they may offer an accessible and scalable way to improve learning outcomes for students using LLMs.

1.2 Tasks

This study's primary goal is to evaluate the effectiveness of prompt-based techniques in reducing hallucinations in Large Language Models (LLMs)**,** with a focus on scenarios where students use these models as learning aids. Instead of relying on model retraining or external tools, this work investigates whether LLM response reliability can be improved using prompt engineering alone.

To this end, the following research questions are investigated:

> **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 these prompting techniques vary across different subject areas?*
>
> **RQ3**: *How consistent are the predictions of each method across repeated runs?*

These questions are explored through a quantitative experiment using a standardized high-school level exam (ENEM 2024) with multiple-choice questions across different subjects. This setup allows for automatic LLM response evaluation and comparison across methods.

The main outcome of this work is a reproducible evaluation framework implemented in Jupyter Notebooks. This framework loads the questions dataset and applies each prompting strategy using the OpenAI API. It then collects results across multiple test runs, and calculates key performance metrics such as accuracy, subject-wise performance, and consistency. All code, prompt templates, and evaluation logic were designed with modularity and transparency in mind, making it straightforward to extend the approach to other datasets or prompting techniques in future studies.

By providing empirical insights and a practical testing pipeline, this thesis aims to contribute to ongoing efforts in making LLM-assisted learning more trustworthy**.**

2 Methodology

2.1 Ideal Solution

An evaluation of prompting techniques for Large Language Models (LLMs) requires a controlled and repeatable experimental setup that allows for fair comparisons across methods. In an ideal scenario, the following conditions would be met:

Fully deterministic LLM behaviour
Each prompt would yield a predictable and reproducible response, eliminating randomness and isolating the effect of prompt structure alone.

Access to model internals
Evaluators would have control over the underlying LLM architecture, token-level probabilities, and intermediate reasoning states, allowing for in-depth analysis.

Large-scale, domain-specific benchmark datasets
Evaluation would be conducted on diverse, verified questions across multiple disciplines, ideally with additional information such as reasoning steps and difficulty levels.

Automated multi-pass evaluation
The system would support large-scale, automated prompting with batch processing, minimal API latency, and robust error handling.

Statistical power and interpretability
Sufficiently large numbers of prompt-response pairs would be generated per method to ensure statistical confidence, and the analysis would include meaningful performance metrics beyond accuracy, such as consistency and reasoning quality.

In practice, evaluating LLM behavior under these ideal conditions is limited by several factors, including API access constrains and computational cost. Therefore, this study adopts a practical approximation of the ideal solution by combining well-defined prompting templates, a standardized evaluation dataset, and repeated test runs. Response consistency was estimated with results from multiple test runs, and the accuracy difference between prompting techniques was validated using statistical tests.

The goal of this setup is not to eliminate uncertainty entirely, but to systematically control for variability while enabling reproducibility and transparent comparison across Chain-of-Thought, Chain-of-Verification, and Self-Refine prompting strategies.

2.2 Requirements

The evaluation metrics and prompt template structures were based on the goals of the selected methods and the prior literature. Accuracy, subject-specific performance, and consistency were chosen to capture performance, domain sensitivity, and stability, respectively, aligning with how prior work evaluated CoT [6], CoVe [9], and Self-Refine [10].

Structured prompt templates were created for each method for consistency across test runs while enabling method-specific logic such as verification or refinement loops.

2.3 Research Questions

This research is guided by the following three questions:

RQ1: *How does the application of CoT, CoVe and Self-Refine affect LLM response accuracy on Brazilian University Entrance Exams?*

RQ2: *How does the effectiveness of CoT, CoVe and Self-Refine vary across different subject areas?*

RQ3: *How consistent are the predictions of Chain-of-Thought, Chain-of-Verification, and Self-Refine methods across repeated runs?*

Each prompting method was implemented as a reusable prompt template and applied to the same multiple-choice question set across multiple runs. The evaluation strategy is based on three core criteria:

Accuracy
The proportion of correct responses produced by each method across all questions.

Subject-specific Performance
The accuracy and consistency of each method when broken down by academic subject (Mathematics, Human Sciences, Languages, and Natural Sciences).

Consistency
The ability of a method to produce the same answer for the same question across repeated test runs.

Together, these analyses provide a comprehensive answer to the research questions. They help identify which technique produces the most accurate, stable, and subject-adaptable results, highlighting the most promising strategy for enhancing reliability in LLM-assisted educational tasks.

2.4 Prompting Techniques

To ensure consistency and control over input formatting, custom templates were created for each of the three prompting methods under evaluation.

2.4.1 Chain-of-Thought (CoT)

Chain-of-Thought (CoT) is a prompting technique designed to improve the reasoning capabilities of LLMs by encouraging them to produce step-by-step explanations before arriving at a final answer. Rather than directly outputting a single response, CoT prompts guide the model through intermediate reasoning steps that mimic human problem-solving behavior. As reported in *Chain-of-Thought Prompting Elicits Reasoning in Large Language*

Models [6] , this structured form of output has shown to significantly improve performance on tasks requiring logical inference, arithmetic, and multi-step analysis.

In the CoT approach, the model follows a three-step process:

1. **Instructions**
   The model is introduced to the task through a 3-shot prompt containing example ENEM questions and detailed explanations. It is then presented with the question it should solve by following the examples.

2. **Generate reasoning**
   The model is expected to produce a chain of reasoning that explains the thought process leading to its answer.

3. **Explicit Final Answer**
   After the reasoning steps, the model is prompted to select and declare the correct multiple-choice alternative.

CoT was implemented using a Python prompt template tailored for ENEM-style multiple-choice questions. Each prompt included three examples taken from *Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams* [11], where the authors evaluated GPT-3.5 and GPT-4 models on ENEM exams. These examples were selected from the ENEM 2022 exam and span across three subject areas: Languages**,** Human Sciences**,** and Mathematics. Each example is formatted to reflect the CoT structure: the question is followed by the answer alternatives, a detailed explanation that analyzes the options, and a final line that explicitly states the correct alternative in a structure format. The explanations were derived from expert teacher discussions and public exam commentary resources, as also documented in the paper [11] and shown in Figure 2. Appendix B presents the original questions and explanations in Portuguese.

In the prompt template, the 3-shot block is followed by the question under evaluation, inserted dynamically during each test run. The model is instructed to produce a full reasoning chain and identify the final answer at the end. Each response was stored with metadata including the selected alternative and the full generated reasoning chain.

Example 3:

Question:
A couple is planning to build a swimming pool on their farm, shaped like a rectangular cuboid with a capacity of 90,000 liters of water. They hired a construction company that presented five designs with different internal combinations of depth, width, and length. The pool will have an internal lining on its walls and bottom using the same ceramic material, and the couple will choose the design that requires the smallest lining area. The internal dimensions of depth, width, and length, respectively, for each project are:
Project I: 1.8 m, 2.0 m, and 25.0 m
Project II: 2.0 m, 5.0 m, and 9.0 m
Project III: 1.0 m, 6.0 m, and 15.0 m
Project IV: 1.5 m, 15.0 m, and 4.0 m
Project V: 2.5 m, 3.0 m, and 12.0 m

The project the couple should choose is:

Options:
A) I.
B) II.
C) III.
D) IV.
E) V.

Explanation:
We must calculate the area of the four lateral faces and the bottom area (pool floor), then sum these areas to get the total lining area. So, calculating the lining area for each project, we have:
Project I: $A = 2 \times 25 + 2 \times 1.8 \times (2 + 25) = 147.2$
Project II: $A = 9 \times 5 + 2 \times 2 \times (9 + 5) = 101$
Project III: $A = 15 \times 6 + 2 \times 1 \times (15 + 6) = 132$
Project IV: $A = 4 \times 15 + 2 \times 1.5 \times (15 + 4) = 117$
Project V: $A = 3 \times 12 + 2 \times 2.5 \times (3 + 12) = 111$
Therefore, the project with the smallest lining area is Project II, so the correct answer is B.

Final answer: B

Figure 2: One of the three questions selected from the ENEM 2022 exam as few-shot examples in the Chain-of-Thought template.

2.4.2 Chain-of-Verification (CoVe)

Chain-of-Verification (CoVe) is a prompting method developed to reduce hallucinations in LLMs by having the model verify its own outputs. Introduced in *Chain-of-Verification Reduces Hallucination in Large Language Models* [9], CoVe operates on the assumption that LLMs can critically analyse and improve their responses when prompted appropriately. Rather than relying on a single pass, CoVe decomposes the generation process into separate verification steps.

The CoVe process involves four sequential stages:

1. **Generate a Baseline Response**
   An initial draft is generated using the Chain-of-Thought template.

2. **Plan Verification Questions**
   The model reflects on the baseline response and produces verification questions to check for factual correctness or reasoning steps.

3. **Execute Verifications**
   Each verification question is answered in isolation, preventing the model from being biased by its own previous answers.

4. **Generate Final Verified Response**
   Using the original question, the baseline response, and the results of the verifications, the model revises and outputs a final, refined answer.

In this study, CoVe was implemented as a three-step pipeline. All steps are executed inside an iterative loop that repeats the CoVe pipeline over the ENEM questions.

The process begins by generating a baseline response using the 3-shot Chain-of-Thought template described in the CoT section. The second step generates verification questions: the prompt instructs the LLM to reflect on its previous answer and identify key factual or logical claims worth verifying. To ensure a robust verification process, each question is sent to the LLM in isolation, without providing the baseline response in the context: this emulates the *factored* verification setup, which was shown to prevent the model from repeating its hallucinations [9]. Finally, a full prompt that includes the original question, alternatives, the baseline answer, and the verification Q&A pairs is generated. The LLM is then asked to synthesize this information into a revised, fully verified response. This final step mimics the *Factor + Revise* variant, which the original authors found to yield the highest factual accuracy [9].

2.4.3 Self-Refine

Self-Refine is an iterative prompting technique in which a language model improves its own output through cycles of self-critique and revision. Instead of producing a single-shot response, the model first generates an answer, then evaluates it by generating critical feedback, and finally uses that feedback to revise and refine its response. This method has shown to improve LLM performance across diverse tasks, including reasoning, coding, and factual QA [10].

The implementation consisted of three steps:

1. **Initial Answer**
   The model answers the ENEM question using the Chain-of-Thought template, producing an initial answer.

2. **Feedback**
   The model receives a second prompt asking it to critically analyze its own response, pointing out factual mistakes, logical inconsistencies, or poor reasoning.

3. **Refinement**
   The model receives a third prompt to revise its answer taking the generated feedback into account. It is instructed to re-express the reasoning and conclude with a final answer in the specified format.

This feedback–refinement loop is repeated until the selected multiple-choice answer remains unchanged between iterations, or until a maximum of 10 iterations is reached. Throughout this process, all iterations, responses, feedbacks, and selected answers are stored in a structured format, including metadata like subject and answer trace.

2.5 Dataset

This study used the open-access Maritaca AI ENEM dataset [11]. It contains multiple-choice questions from the 2024 *Exame Nacional do Ensino Médio* (ENEM), Brazil's national standardized exam used for university admissions. A total of 180 questions were used, covering four official subject areas:

- Questions 1–45: *Linguagens, Códigos e suas Tecnologias* (Languages)
- Questions 46–90: *Ciências Humanas e suas Tecnologias* (Human Sciences)
- Questions 91–135: *Ciências da Natureza e suas Tecnologias* (Natural Sciences)
- Questions 136–180: *Matemática e suas Tecnologias* (Mathematics)

Each entry included the question, the multiple-choice alternatives and a ground-truth label.

2.6 Tests

A testing pipeline was developed so that it systematically applies each prompting technique to the full ENEM 2024 dataset. For each method there is a total of 40 responses per question.

For each prompt-response pair, the following data points were collected:

- The full model output, including reasoning and answer,
- The extracted answer letter (A–E),
- Whether the prediction was correct,
- The subject label,
- Additional metadata depending on the method:
    - CoVe: Verification questions and answers
    - Self-Refine: Iterative responses, feedback trace and intermediate answers

2.7 Tools

All experiments, data processing, and analyses were conducted using openly available tools and Python libraries. The following technologies were used throughout the development and evaluation process:

Jupyter Notebook
Provided an interactive environment for developing, testing, and documenting the experimental pipeline. It facilitated reproducibility and allowed step-by-step analysis of intermediate outputs, errors, and statistical results.

OpenAI API
Used to generate model responses via the GPT-3.5 Turbo language model. The API enabled

programmatic submission of prompts and retrieval of outputs across multiple runs and prompting templates.

pandas
Served as the primary tool for data manipulation, grouping, filtering, and cleaning of results. It was used extensively to structure run outputs, calculate performance metrics, and organize data for visualization.

matplotlib and **seaborn**
These libraries were used for generating all plots and visualizations in the analysis. Bar plots, box plots, and significance markers were created to communicate results in a clear and informative way.

numpy
Provided support for numerical operations, including array-based computations and statistical aggregation.

scipy
Used for statistical testing, including Shapiro–Wilk tests for normality, Levene's test for variance homogeneity, Kruskal–Wallis tests for group comparison, and McNemar's tests for paired binary outcomes.

3 Results

To compare the effectiveness of the prompting techniques, a quantitative analysis was performed based on three key metrics: overall accuracy, subject-wise accuracy, and consistency across runs. All analyses were conducted on the cleaned datasets generated for each of the 40 test runs.

3.1 General Accuracy

A general comparison of accuracy distributions for each method was conducted to address the **Research Question 1**:

RQ1: *How does the application of CoT, CoVe and Self-Refine affect LLM response accuracy on Brazilian University Entrance Exams?*

To measure overall accuracy, the results from all runs of each method were combined into a single DataFrame. Data was first grouped by method and run to compute per-run accuracy and results were then aggregated to calculate the mean and standard deviation.

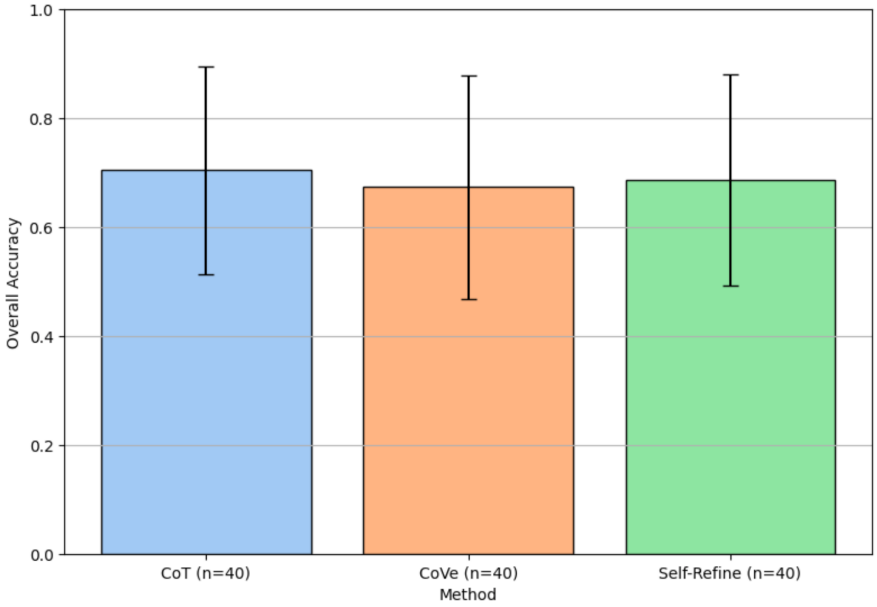

Figure 3: Overall accuracy and standard deviation by method

The following table shows the average accuracy across all ENEM questions for each technique, along with the standard deviation:

Method	**Accuracy**	**Runs**
CoT	0.704 ± 0.190	40
CoVe	0.673 ± 0.206	40
Self-Refine	0.686 ± 0.193	40

Table 1: Mean accuracy and standard deviation of CoT, CoVe and Self-Refine across test runs

The next step was to assess whether the differences in overall accuracy between the three methods were statistically significant. First it was tested whether the conditions for parametric analysis were met:

- **Normality (Shapiro–Wilk test):**
  - CoT: W = 0.8983, $p < 0.0001$
  - CoVe: W = 0.9108, $p < 0.0001$
  - Self-Refine: W = 0.8929, $p < 0.0001$

  All three distributions showed statistically significant deviation from normality ($p < 0.05$), indicating that the normality assumption was violated.

- **Homogeneity of Variances (Levene's test):**
  - Statistic = 1.2022, $p = 0.3014$

  The results of Levene's test indicate that the variance across groups is not significantly different, satisfying the assumption of equal variances.

Given the non-normality of the data, the Kruskal–Wallis H test [12] was applied to compare the distributions of accuracy across the three methods.

- **H_0 (Null Hypothesis):** The distributions of accuracy scores across the three prompting methods are the same.
- **H_1 (Alternative Hypothesis):** At least one method's distribution differs significantly from the others.

- **H statistic**: 3.7021
- ***p* -value**: 0.1570

The Kruskal–Wallis test resulted in a p-value of 0.1570, which is above the conventional threshold of 0.05, so **it is not possible to reject the null hypothesis (H_0).** This suggests that none of the prompting techniques consistently outperformed the others across all evaluation runs. While Chain-of-Thought achieved the highest mean accuracy (70.4%), followed by Self-Refine (68.6%) and Chain-of-Verification (67.3%), these observed differences could be due to random variation.

To assess whether the non-significance could be explained by insufficient sample size, a post-hoc power analysis was conducted using the observed means and standard deviations of the three methods. The effect size, computed as Cohen's $f = 0.079$, reflects a very small between-group difference relative to the within-group variability [13]. Based on this effect size, an α level of 0.05, and a power of 0.80, the estimated required sample size is approximately 1,536 runs per method. This confirms that the current experiment was underpowered to detect such small differences, and that larger-scale testing would be needed to identify statistically significant effects.

Conclusion for RQ1
This finding brings an answer to Research Question 1 (RQ1): while CoT exhibits higher performance, the statistical analysis does not support a definitive conclusion that it is more accurate than CoVe or Self-Refine.

3.2 Subject-Specific Accuracy

A subject-wise comparison of accuracy distributions was conducted to address the **Research Question 2:**

RQ2: *How does the effectiveness of Chain-of-Thought, Chain-of-Verification, and Self-Refine vary across different subject areas?*

First, the accuracy scores were grouped by `method`, `run`, and `subject`, and the mean accuracy was calculated for each unique combination.

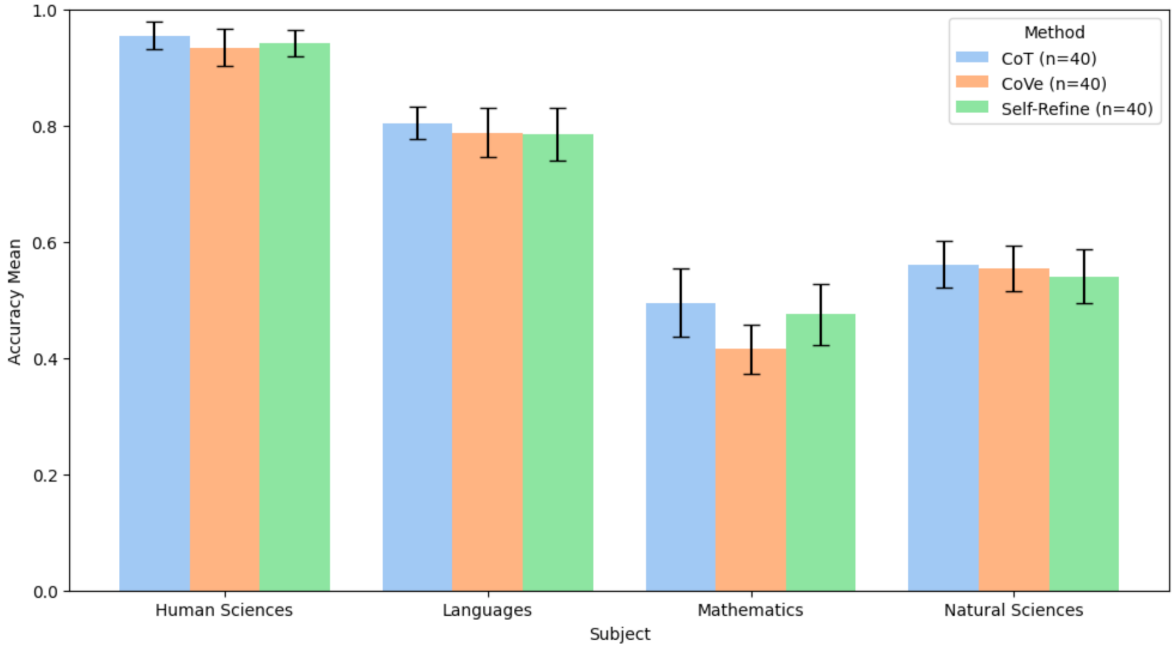

Figure 4: Accuracy mean and standard deviation per method and subject

Considering the violations of the normality assumption in the accuracy distributions, as established in the previous section, the Kruskal–Wallis H test was employed to determine whether the prompting methods differed significantly in accuracy for specific subjects.

- **H_0 (Null Hypothesis):** The distributions of accuracy scores across the three prompting methods are statistically indistinguishable for any subject.

- **H_1 (Alternative Hypothesis):** At least one method differs significantly in accuracy from the others for a given subject.

Statistically significant differences were observed for Mathematics and Human Sciences, indicating that at least one method outperformed the others within these subjects. In both cases, Chain-of-Thought achieved the highest average accuracy. No significant differences were found for Languages and Natural Sciences, suggesting comparable performance across methods for these subjects.

Subject	**CoT Accuracy**	**CoVe Accuracy**	**Self-Refine Accuracy**	**Kruskal-Wallis H**	***p*-value**	**Runs**
Human Sciences	0.955 ± 0.024	0.934 ± 0.031	0.942 ± 0.023	10.425922	0.005446 ***	40
Languages	0.804 ± 0.028	0.788 ± 0.043	0.786 ± 0.046	5.508679	0.06365	40
Mathematics	0.495 ± 0.059	0.415 ± 0.042	0.475 ± 0.052	37.652330	6.667e-09 ***	40
Natural Sciences	0.561 ± 0.041	0.554 ± 0.039	0.540 ± 0.046	5.743476	0.0566	40

Table 2: Mean accuracy ± standard deviation per subject and method across 40 runs. Kruskal–Wallis H test checks for accuracy differences between methods. Significance: *** $p < 0.01$.

For **Mathematics** and **Human Sciences**, the p-values are both below the significance threshold of 0.05. Therefore, **the null hypothesis (H_0) is rejected** for Mathematics and Human Sciences, as there are statistically significant differences in accuracy between the prompting methods. In both of these subjects, **Chain-of-Thought achieved the highest mean accuracy** (Mathematics: 0.495, Human Sciences: 0.955), indicating it outperformed both Chain-of-Verification and Self-Refine.

For **Languages** ($p > 0.06$) and **Natural Sciences** ($p > 0.05$), the p-values are above the 0.05 threshold. Therefore, **it is not possible to reject the null hypothesis (H_0)** for these subjects, indicating that there is no statistically significant difference in accuracy between the three prompting methods.

Conclusion for RQ2
These findings bring an answer to Research Question 2 (RQ2): there was a significant difference between the methods' performance in Mathematics and Human Sciences, with Chain-of-Thought offering the most benefit in both cases. In subjects like Languages and Natural Sciences, all methods perform similarly.

3.2.1 Insights on Feedback Effects

A closer examination of the subject-wise accuracy gains reveals insights about the effectiveness of feedback-based techniques in this context. While Self-Refine and CoVe are designed to improve upon initial answers through iterative reasoning or verification, results show that this is not always the case.

A test was performed to assess what would happen if both methods had always stuck with their initial answer: the output produced using the Chain-of-Thought template, before any feedback or revision. Surprisingly, in several cases, this baseline answer proved to be statistically more accurate than the final refined one.

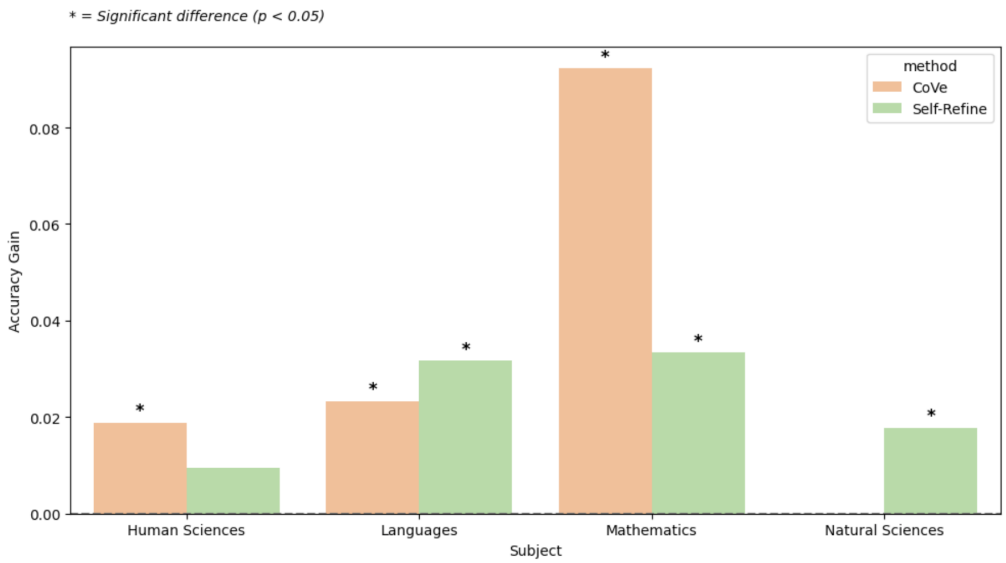

Figure 5: Accuracy gain by sticking with the initial answer

Method	Subject	Normality p	Test	*p*-Value	Runs
CoVe	Human Sciences	0.000210	one-sided Wilcoxon	1.05×10^{-4} ***	40
	Languages	0.074402	one-sided paired t-test	2.74×10^{-5} ***	40
	Mathematics	0.195119	one-sided paired t-test	3.43×10^{-13} ***	40
	Natural Sciences	0.017626	one-sided Wilcoxon	3.44×10^{-1}	40
Self-Refine	Human Sciences	0.014433	one-sided Wilcoxon	5.98×10^{-2}	40
	Languages	0.017444	one-sided Wilcoxon	9.80×10^{-5} ***	40
	Mathematics	0.055846	one-sided paired t-test	1.17×10^{-4} ***	40
	Natural Sciences	0.081444	one-sided paired t-test	1.08×10^{-2} **	40

Table 3: Accuracy gain when sticking to the first answer instead of refining. Statistical significance is based on one-sided paired t-tests or Wilcoxon tests depending on normality.
Significance notation: ** $p < 0.05$; *** $p < 0.01$.

CoVe

For CoVe, the final refinement underperformed the initial CoT answer in three out of four subjects (Human Sciences, Languages, and Mathematics) with statistically significant drops in accuracy ($p < 0.001$ for Mathematics, for example). The only exception was in Natural Sciences, where the difference was not statistically significant. This pattern suggests that CoVe's verification mechanism may sometimes introduce errors or reinforce uncertainties rather than resolving them, possibly due to flawed verification questions or overly cautious revisions.

Self-Refine

Self-Refine showed a consistent trend of accuracy decrease after refinement in most subject areas. In Languages**,** Mathematics**,** and Natural Sciences, the final answer generated after the feedback–refinement loop was significantly worse than the initial response generated using the Chain-of-Thought template ($p < 0.001$ for both Languages and Mathematics; $p = 0.01$ for Natural Sciences). These results suggest that, rather than improving the original answer, the iterative feedback mechanism of Self-Refine often introduced noise or instability that led to poorer outcomes.

Conclusion

Overall, these results highlight a critical insight**: answer refinement do not always lead to more accurate answers**. In fact, for certain subjects and methods, the best result was already achieved in the first answer**,** raising questions about when and how refinement should be applied. It suggests that a hybrid strategy, where refinement is selectively applied based on initial confidence or answer quality, may be a more effective approach in educational tasks.

3.3 Consistency

Large Language Models are stochastic, meaning the same prompt can lead to different outputs across runs. This section addresses **Research Question 3:**

RQ3: *How consistent are the predictions of Chain-of-Thought, Chain-of-Verification, and Self-Refine methods across repeated runs?*

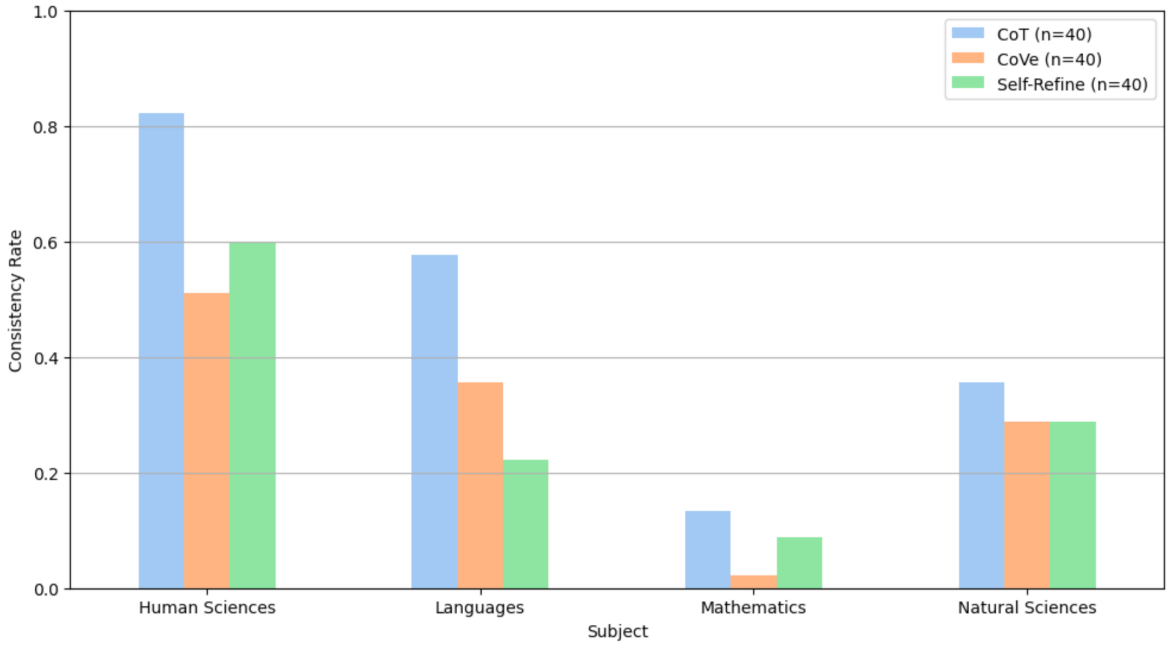

Figure 6: Answer consistency per method and subject

For this analysis, a prediction was marked as consistent if the model produced the same final answer for the same question across all 40 test runs.

Method	Consistency Rate	Runs
CoT	47.2%	40
CoVe	29.4%	40
Self-Refine	30.0%	40

Table 4: Overall consistency rate per method across test runs

These results indicate that Chain-of-Thought produces the most stable predictions, followed by Self-Refine and CoVe.

To compare consistency between prompting methods, the McNemar's test [14] was selected. The test is designed for paired binary data, and in this study, each method's prediction was classified as either consistent or inconsistent across runs for a given question. This creates matched observations suitable for McNemar's test.

The test assesses whether the frequency of discordant outcomes differs significantly between two methods, for example, whether one is more likely to produce consistent answers than the other.

- **H_0 (Null Hypothesis):** The consistency proportions of the two methods being compared are equal (no difference in consistency).
- **H_1 (Alternative Hypothesis):** The consistency proportions differ between the two methods.

Comparison	***p* -value**
CoT vs CoVe	< 0.0001 ***
CoT vs Self-Refine	< 0.0001 ***
CoVe vs Self-Refine	1.0000

Table 5: McNemar's test results for answer consistency. Significance notation: *** $p < 0.01$

The results show that CoT is significantly more consistent than both CoVe and Self-Refine ($p < 0.01$), while there is no statistically significant difference between CoVe and Self-Refine ($p = 1.0$).

Subject	**Comparison**	***p* -value**
Human Sciences	CoT vs CoVe	0.0001 ***
	CoT vs Self-Refine	0.0020 ***
	CoVe vs Self-Refine	0.3438
Languages	CoT vs CoVe	0.0213 **
	CoT vs Self-Refine	0.0004 ***
	CoVe vs Self-Refine	0.1460
Mathematics	All comparisons	> 0.06
Natural Sciences	All comparisons	> 0.3

Table 6: McNemar's test results for answer consistency per subject.
Significance notation: ** $p < 0.05$; *** $p < 0.01$.

Based on the test results, **the null hypothesis (H_0) is rejected** for comparisons involving Chain-of-Thought (CoT) versus both Chain-of-Verification (CoVe) and Self-Refine, indicating that **CoT is significantly more consistent** overall ($p < 0.01$). However, **the null hypothesis (H_0) is not rejected** for CoVe versus Self-Refine ($p = 1.0$), suggesting no significant difference in their consistency.

At the subject level, CoT also shows significantly higher consistency than both CoVe and Self-Refine in **Human Sciences** and **Languages**, but no significant differences were observed in **Mathematics** and **Natural Sciences**. These results support the alternative hypothesis (H_1) for Human Sciences and Languages domains.

Conclusion for RQ3
These results answer Research Question 3 (RQ3): Chain-of-Thought produces significantly more consistent predictions than both Chain-of-Verification and Self-Refine, both overall and within specific subject areas like Human Sciences and Languages.

4 Discussion

4.1 Potentials

A key strength of this study was the use of multiple test runs per method. Repeating the experiment 40 times helped reveal not only which methods were more accurate on average, but also which ones gave stable answers when faced with the same question multiple times. Measuring consistency is especially important when working with stochastic models like GPT-3.5 Turbo, where the same input can produce different outputs.

Another important part of the methodology was the use of prompting templates. Each template was tailored to follow the logic of the corresponding method. This preserved the unique reasoning structure of each method and ensured a fair comparison. Templates were applied consistently across all questions and test runs, reducing bias caused by formatting or instruction differences.

The ENEM 2024 dataset used in this study also provided a realistic and relevant test scenario. These are real multiple-choice questions from a national university entrance exam in Brazil, covering subjects like Math, Languages, and Sciences. The dataset was already labelled with correct answers and followed a fixed structure, which allowed for automatic scoring and subject-level analysis without manual grading.

Together, these methodological choices helped make the results reproducible, fair, and applicable to real-world student use cases.

4.2 Limitations

This study has several limitations that should be considered when interpreting the results. First, all experiments were conducted using GPT-3.5 Turbo**.** While this is a powerful model, newer or larger models like GPT-4 might behave differently. The results may not fully generalize to other LLMs with different training data or architectures.

Second, the evaluation was based on a single dataset, the ENEM 2024 Brazilian university entrance exam. Although it covers a variety of subjects, it is limited to multiple-choice questions, which have a fixed structure and predefined answer options. Therefore, the findings might not apply to open-ended questions, essay-style responses, or other formats.

Finally, computational and API cost constraints limited the scale of the experiment. Only 180 questions were tested, and each method was run 40 times. A post-hoc power analysis based on the observed means and standard deviations indicates that detecting a statistically significant difference between methods with 80% power would require approximately 1,536 runs per method. This highlights the difficulty of identifying small but consistent performance differences with limited resources.

These limitations suggest that while the results are useful for understanding LLM performance in this specific setting, further research is needed to explore how these prompting techniques perform across other models, datasets, and question types.

4.3 Possible Improvements

Test with Multiple Models
Using other models, such as GPT-4, Claude, or LLaMA would help determine whether the findings hold across architectures and levels of capability. This could reveal whether certain prompting techniques are more model-dependent.

Include More Diverse Datasets
Expanding the evaluation beyond ENEM 2024 to include other standardized exams or open-ended question formats would test how well the methods generalize.

Analyze Reasoning Quality
While this study focused on accuracy, a qualitative analysis of the reasoning chains, feedback messages, or verification steps could offer deeper insight into how and why methods succeed or fail.

Adaptive Prompting Strategies
One limitation of methods like Self-Refine is that they apply refinement to all questions, even when it is not needed. Future work could explore hybrid or adaptive strategies, where refinement is only triggered if the model's confidence is low or if the reasoning shows signs of error.

Larger-Scale Experiments
Running experiments on larger datasets or increasing the number of repeated runs per question could improve statistical power and help reduce uncertainty. This would allow for more robust conclusions and better handling of model variance.

4.4 Generalizability of the Solution

The solution presented in this study is modular and adaptable**,** making it possible to apply it in other contexts beyond the ENEM 2024 dataset with small adjustments. Since all prompting techniques were implemented without relying on model fine-tuning or external tools, they can be reused with any LLM accessible via API.

The methodology can generalize well to other multiple-choice exams, especially those with a clear structure and labeled answers, such as SATs, or national exams from other countries. With small adjustments to the prompt templates (e.g., using domain-specific examples), the same approach could be applied to different subject areas or educational levels.

However, the current setup is limited when it comes to open-ended tasks, such as writing essay, programming, and answering open questions. In those cases, different evaluation methods would be required, including qualitative scoring or human feedback.

In summary, the solution is general enough for structured QA tasks across different exams and domains but would require adaptation for more complex applications.

4.5 Takeaways

This study shows that it is possible to design an evaluation framework for LLM prompting methods using only publicly available tools and datasets. One key takeaway is the importance of modular design: by separating data loading, prompting, testing, and analysis into reusable components, the entire pipeline became easier to debug, extend, and scale.

The use of multiple runs per method was also essential. It revealed how inconsistent LLMs can be and why performance should not be measured with a single test. This is an important insight for researchers and developers working with LLMs: measuring only accuracy without considering variability can be misleading.

Finally, this study shows that LLM experimentation can be practical and reproducible, even for students or small teams. With clear goals, thoughtful design, and systematic testing, it is possible to generate meaningful insights without access to internal model weights or advanced infrastructure.

Bibliography

[1] T. Zhu, K. Zhang und W. Y. Wang, „Embracing AI in Education: Understanding the Surge in Large Language Model Use by Secondary Students," arXiv preprint arXiv:2411.18708, 2024.

[2] J. Freeman, „Provide or punish? Students' views on generative AI in higher education," Higher Education Policy Institute, 2024.

[3] J. Fisher, „Making the most of ChatGPT in the classroom," Carnegie Learning, 16 April 2024. [Online]. Available: https://www.carnegielearning.com/blog/making-the-most-of-chatgpt. [Zugriff am 10 May 2025].

[4] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Qian, X. Xiao, Q. Guo, Q. Liu, X. Xue und Z. Liu, „Survey of hallucination in natural language generation," *ACM computing surveys 55,* pp. 1-38, March 2023.

[5] J. Maynez, S. Narayan, B. Bohnet und R. McDonald, „On Faithfulness and Factuality in Abstractive Summarization," arXiv preprint arXiv:2005.00661, 2020.

[6] 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.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, S. Riedel, D. Kiela und L. Zettlemoyer, „Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS),* 2020.

[8] B. Lester, R. Al-Rfou und N. Constant, „The Power of Scale for Parameter-Efficient Prompt Tuning," arXiv preprint arXiv:2104.08691, 2021.

[9] 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.

[10] 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.

[11] 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.

[12] W. H. Kruskal und W. A. Wallis, „Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association,* Bd. 47, Nr. 260, pp. 583-621, 1952.

[13] J. Cohen, „Statistical power analysis for the behavioral sciences," routledge, 2013.

[14] Q. McNemar, „Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika,* p. 153–157, 1947.

[15] B. Peng, C. Li, P. He, M. Galley und J. Gao, „Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback," in *Proceedings of the Seventh Workshop on e-Commerce and NLP*, 2023.

[16] J. Ye, N. Xu, Y. Wang, J. Zhou, Q. Zhang, T. Gui und X. Huang, „LLM-DA: Data Augmentation via Large Language Models for Few-Shot Named Entity Recognition," arXiv preprint arXiv:2402.14568, 2024.

[17] B. Ding, C. Qin, R. Zhao, T. Luo, X. Li, G. Chen, W. Xia, J. Hu, A. T. Luu und S. Joty, „Data Augmentation using Large Language Models: Data Perspectives, Learning Paradigms and Challenges," arXiv preprint arXiv:2403.02990, 2024.

[18] Z. Zhang, Y. Fan, Y. Wang, Z. Wang und X. Wan, „LLM-DA++: A Unified Framework of Data Augmentation Using Large Language Models," arXiv preprint arXiv:2403.12345, 2024.

[19] Y. Zhou, S. Shan, H. Wei, Z. Zhao und W. Feng, „PGA-SciRE: Harnessing LLM on Data Augmentation for Enhancing Scientific Relation Extraction," arXiv preprint arXiv:2405.20787, 2024.

[20] S. Z. Qiu Q, „A nonuniform weighted loss function for imbalanced image classification," in *Proceedings of the 2018 international conference on image and graphics processing*, 2018.

[21] T. Vu, M. Iyyer, X. Wang, N. Constant, J. Wei, J. Wei, C. Tar, Y.-H. Sung, D. Zhou, Q. Le und T. Luong, „Freshllms: Refreshing large language models with search engine augmentation," arXiv preprint arXiv:2310.03214, 2023.

[22] H. Cao, Z. An, J. Feng, K. Xu, L. Chen und D. Zhao, „A Step Closer to Comprehensive Answers: Constrained Multi-Stage Question Decomposition with Large Language Models," arXiv preprint arXiv:2311.07491, 2023.

[23] H. Kang, J. Ni und H. Yao, „Ever: Mitigating hallucination in large language models through real-time verification and rectification," arXiv preprint arXiv:2311.09114, 2023.

[24] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Y. Zhao, N. Lao, H. Lee, D.-C. Juan und K. Guu, „Rarr: Researching and revising what language models say, using language models," arXiv preprint arXiv:2210.08726, 2022.

[25] C. Si, Z. Gan, Z. Yang, S. Wang, J. Wang, J. Boyd-Graber und L. Wang, „Prompting GPT-3 to be reliable," arXiv preprint arXiv:2210.09150, 2022.

[26] N. Mündler, J. He, S. Jenko und M. Vechev, „Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation," arXiv preprint arXiv:2305.15852, 2023.

[27] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii und P. Fung, „Towards mitigating hallucination in large language models via self-reflection," arXiv preprint arXiv:2310.06271, 2023.

[28] J. N. Yan, T. Liu, J. T. Chiu, J. Shen, Z. Qin, Y. Yu, Y. Zhao, C. Lakshmanan, Y. Kurzion, A. M. Rush, J. Liu und M. Bendersky, „Predicting text preference via structured comparative reasoning," arXiv preprint arXiv:2311.08390, 2023.

[29] W. Liu, G. Li, K. Zhang, B. Du, Q. Chen, X. Hu, H. Xu, J. Chen und J. Wu, „Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models," arXiv preprint arXiv:2311.09214, 2023.

[30] Y. Chen, K. Sikka, M. Cogswell, H. Li, D. Batra und D. Parikh, „Dress: Instructing large vision-language models to align and interact with humans via natural language feedback," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[31] M. Elaraby, M. Lu, J. Dunn, X. Zhang, Y. Wang, S. Liu, P. Tian, Y. Wang und Y. Wang, „Halo: Estimation and reduction of hallucinations in open-source weak large language models," arXiv preprint arXiv:2308.11764, 2023.

[32] K. Tian, E. Mitchell, H. Yao, C. D. Manning und C. Finn, „Fine-tuning language models for factuality," in *The Twelfth International Conference on Learning Representations*, 2023.

[33] H. Zhang, S. Diao, Y. Lin, Y. R. Fung, Q. Lian, X. Wang, Y. Chen, H. Ji und T. Zhang, „R-tuning: Instructing large language models to say 'I don't know," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.

[34] Y. Qiu, V. Embar, S. B. Cohen und B. Han, „Think while you write: Hypothesis verification promotes faithful knowledge-to-text generation," arXiv preprint arXiv:2311.09467, 2023.

[35] D. Lei, Y. Li, M. (. Hu, M. Wang, V. Yun, E. Ching und E. Kamal, „hain of natural language inference for reducing large language model ungrounded hallucinations," arXiv preprint arXiv:2310.03951, 2023.

[36] D. Cheng, S. Huang, J. Bi, Y. Zhan, J. Liu, Y. Wang, H. Sun, F. Wei, D. Deng und Q. Zhang, „Uprise: Universal prompt retrieval for improving zero-shot evaluation," arXiv preprint arXiv:2303.08518, 2023.

[37] E. Jones, H. Palangi, C. Simões, V. Chandrasekaran, S. Mukherjee, A. Mitra, A. Awadallah und E. Kamar, „Teaching language models to hallucinate less with synthetic tasks," arXiv preprint arXiv:2310.06827, 2023.

[38] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. R. Glass und P. He, „Dola: Decoding by contrasting layers improves factuality in large language models," arXiv preprint arXiv:2309.03883, 2023.

[39] K. Li, O. Patel, F. Viégas, H. Pfister und M. Wattenberg, „Inference-time intervention: Eliciting truthful answers from a language model," *Advances in Neural Information Processing Systems,* Bd. 36, 2023.

[40] F. F. Bayat, K. Qian, B. Han, Y. Sang, A. Belyi, S. Khorshidi, F. Wu, I. Ilyas und Y. Li, „Fleek: Factual error detection and correction with evidence retrieved from external knowledge," arXiv preprint arXiv:2310.17119, 2023.

Figures

Tables

Documentation table of AI-based tools

AI-based tools	**Intended use**	**Prompt, source, page, paragraph...**
ChatGPT (4o)	Translation of a few-shot example to English	"Translate this to English keeping the same format: ..." (Example 1 used in Chain-of-Thought template)
ChatGPT (4o)	Spell-check	“Check this document for spell errors and grammar mistakes” (Entire document)

Appendix

A. All code used in this thesis, including data preprocessing, prompting templates, and evaluation scripts, is publicly available at: https://github.com/rafaelasantana/bachelor-thesis

B. Few-shot examples used in the Chain-of-Thought template (in Portuguese)

Exemplo 1:

Pergunta:
Urgência emocional. Se tudo é para ontem, se a vida engata uma primeira e sai em disparada, se não há mais tempo para paradas estratégicas, caímos fatalmente no vício de querer que os amores sejam igualmente resolvidos num átimo de segundo. Temos pressa para ouvir "eu te amo". Não vemos a hora de que fiquem estabelecidas as regras de convívio: somos namorados, ficantes, casados, amantes? Urgência emocional. Uma cilada. Associamos diversas palavras ao AMOR: paixão, romance, sexo, adrenalina, palpitação. Esquecemos, no entanto, da palavra que viabiliza esse sentimento: "paciência". Amor sem paciência não vinga. Amor não pode ser mastigado e engolido com emergência, com fome desesperada. É uma refeição que pode durar uma vida. MEDEIROS, M. Disponível em: http://porumavidasimples.blogspot.com.br. Acesso em: 20 ago. 2017 (adaptado).

Nesse texto de opinião, as marcas linguísticas revelam uma situação distensa e de pouca formalidade, o que se evidencia pelo(a)

Opções:
A) A impessoalização ao longo do texto, com em: "se não há mais tempo".
B) A construção de uma atmosfera de urgência, em palavras como: "pressa".
C) A repetição de uma determinada estrutura sintática, como em: "Se tudo é para ontem".
D) O ênfase no emprego de hipérboles, como em: "uma reflexão que pode durar uma vida".
E) O emprego de metáforas, como em: "a vida engata uma primeira e sai em disparada".

Explicação:
O texto é escrito em uma linguagem leve, ágil, e de pouca formalidade. Além disso, possui figuras de linguagem, como metáforas e hipérboles, que não são excludentes. Em uma análise sequencial das alternativas, daria para afirmar que D. e E. estão corretas. Entretanto, observando em detalhes, nota-se que a expressão "emprego de metáforas" mostra ser mais adequada do que "ênfase no emprego da hipérbole", visto que, para afirmarmos que o uso de hipérboles foi enfatizado, a figura de linguagem deveria ter aparecido mais vezes. Isso torna a alternativa E. mais provável de ser CORRETA. Além disso, impessoalização não deve ser apontada como marca de pouca formalidade. Existe também uma atmosfera de urgência, mas que é criticada no texto que destaca a importância da paciência e não da pressa. Por fim, a estrutura sintática não é repetida sistematicamente ao longo do texto.

Resposta final: E

Exemplo 2:

Pergunta:
Sempre que a relevância do discurso entra em jogo, a questão torna-se política por definição, pois é o discurso que faz do homem um ser político. E tudo que os homens fazem, sabem ou experimentam só tem sentido na medida em que pode ser discutido. Haverá, talvez, verdades que ficam além da linguagem e que podem ser de grande relevância para o homem no singular, isto é, para o homem que, seja o que for, não é um ser político. Mas homens no plural, isto é, os homens que vivem e se movem e agem neste mundo, só podem experimentar o significado das coisas por poderem falar e ser inteligíveis entre si e consigo mesmos. ARENDT, H. A condição humana. Rio de Janeiro: Forense Universitária, 2004.

No trecho, a filósofa Hannah Arendt mostra a importância da linguagem no processo de

Opções:
A) entendimento da cultura.
B) aumento da criatividade.
C) percepção da individualidade.
D) melhoria da técnica.
E) construção da sociabilidade.

Explicação:
Hannah Arendt defende em sua obra que somos seres políticos, no sentido próprio de vivermos em pólis, em ambiente coletivo e social. E essa sociabilidade só é possível por meio do discurso, da linguagem. Desse modo, podemos concluir que a linguagem se apresenta como uma importante ferramenta para a construção da sociabilidade, e portanto a alternativa E. é a CORRETA. Além disso, não se trata do entendimento da cultura, mas da relação social entre as pessoas dessa cultura. Hannah também não fala sobre aumento de criatividade, tampouco sobre técnica. Por fim, a linguagem é utilizada em algo mais coletivo e social, justamente o oposto da individualidade.

Resposta final: E

Exemplo 3:

Pergunta:
Um casal planeja construir em sua chácara uma piscina com o formato de um paralelepípedo reto retângulo com capacidade para 90 000 L de água. O casal contratou uma empresa de construções que apresentou cinco projetos com diferentes combinações nas dimensões internas de profundidade, largura e comprimento. A piscina a ser construída terá revestimento interno em suas paredes e fundo com uma mesma cerâmica, e o casal irá escolher o projeto que exija a menor área de revestimento. As dimensões internas de profundidade, largura e comprimento, respectivamente, para cada um dos projetos, são: projeto I: 1,8 m, 2,0 m e 25,0 m; projeto II: 2,0 m, 5,0 m e 9,0 m; projeto III: 1,0 m, 6,0 m e 15,0 m; projeto IV: 1,5 m, 15,0 m e 4,0 m; projeto V: 2,5 m, 3,0 m e 12,0 m.

O projeto que o casal deverá escolher será o

Opções:
A) I.
B) II.
C) III.
D) IV.
E) V.

Explicação:
Devemos calcular a área das quatro faces laterais e a área da base inferior (fundo da piscina) e somar essas áreas para obter a área de revestimento. Logo, calculando a área de revestimento de cada projeto, temos: Projeto I: A = 2 x 25 + 2 x 1,8 x (2 + 25) = 147,2; Projeto II: A = 9 x 5 + 2 x 2 x (9 + 5) = 101; Projeto III: A = 15 x 6 + 2 x 1 x (15 + 6) = 132; Projeto IV: A = 4 x 15 + 2 x 1,5 x (15 + 4) = 117; Projeto V: A = 3 x 12 + 2 x 2,5 x (3 + 12) = 111. Logo, o projeto com menor área de revestimento, é o projeto II, portanto a resposta corrreta é B.

Resposta final: B