FACULDADE DE TECNOLOGIA DE PRESIDENTE PRUDENTE
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RAYAN ANZAI

IA – UM ESTUDO SOBRE ESCRITA DE *PROMPTS*

Presidente Prudente – SP

2024

FACULDADE DE TECNOLOGIA DE PRESIDENTE PRUDENTE
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RAYAN ANZAI

IA – UM ESTUDO SOBRE ESCRITA DE *PROMPTS*

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Presidente Prudente, como requisito parcial para obtenção do diploma de Tecnólogo em Análise e Desenvolvimento de sistemas.

Orientadora: Profa. Dra. Ana Carolina Nicolosi da Rocha Gracioso.

Presidente Prudente – SP

2024

RAYAN ANZAI

IA – UM ESTUDO SOBRE ESCRITA DE *PROMPTS*

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Presidente Prudente, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em: 03 de dezembro de 2024.

BANCA EXAMINADORA

_____
Orientadora: Profa. Dra. Ana Carolina N R Gracioso
Faculdade de Tecnologia - Fatec
Presidente Prudente

_____
Profa. Me. Adriane Cavichiolli
Faculdade de Tecnologia - Fatec
Presidente Prudente

_____
Prof. Me. Dione Jonathan Ferrari
Faculdade de Tecnologia - Fatec
Presidente Prudente

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela força, sabedoria e graça que me sustentaram ao longo dessa caminhada.

A minha orientadora, Ana Carolina Nicolosi da Rocha Gracioso, expresso minha profunda gratidão por sua paciência, acolhimento, dedicação e compreensão ao longo de cada etapa deste TCC. Suas palavras e conselhos valiosos foram fundamentais para que eu pudesse concluir mais esse desafio.

Aos meus amigos que estiveram ao meu lado nessa jornada, compartilhando momentos de motivação e apoio, foram essenciais para manter o foco durante esse processo.

E à minha família, obrigado por acreditarem em mim, mesmo nos momentos de dúvida e cansaço. Obrigado por me apoiarem na minha transição de carreira. O suporte de vocês é meu maior tesouro e minha fonte de inspiração para continuar seguindo em frente.

A todos vocês, meu obrigado eterno!

“Só se pode alcançar um grande êxito quando nos mantemos fiéis a nós mesmo.” – Friedrich Nietzsche.

RESUMO

ANZAI, Rayan. **IA – Um estudo sobre escrita de Prompts**. Orientador: Prof. Dra. Ana Carolina Nicolosi da Rocha Gracioso. 2024. 27 f. Trabalho de Conclusão de Curso Tecnologia em Análise e Desenvolvimento de Sistemas - Faculdade de Tecnologia de Presidente Prudente, Presidente Prudente, SP, 2024.

O trabalho abordou a aplicação de técnicas avançadas de engenharia de *prompts* para otimizar a criação automatizada de questões de múltipla escolha no *software* Creator4All, desenvolvido pela empresa Multimídia Educacional. Com o objetivo de aprimorar a precisão das respostas geradas pelo modelo de linguagem ChatGPT, amplamente utilizado na ferramenta, o estudo examinou estratégias de *prompting* como *Zero-shot*, *Few-shot*, *Chain-of-Thought* e *Chain-of-Verification*. Estas técnicas são empregadas para ajustar a capacidade de compreensão e consistência da inteligência artificial (IA) ao responder a questões, especialmente em tarefas complexas que demandam raciocínio lógico. A pesquisa foi realizada em duas fases, envolvendo a criação de 150 questões distribuídas nas disciplinas de Matemática, História e Música, a fim de avaliar a eficácia e precisão do modelo. Na primeira fase, utilizando o ChatGPT 3.5, observou-se que o modelo apresentava alta taxa de acerto em questões teóricas, alcançando 93% de acertos em História. No entanto, o desempenho foi mais limitado em áreas que exigem raciocínio lógico, com uma taxa de acerto de 60% em Matemática e 37% em Música, o que destacou a necessidade de ajustes. Na segunda fase, foram implementadas modificações nos *prompts* e no código do *software*, incluindo a adição do método *Chain-of-Verification*, que verificou a exatidão da resposta antes de apresentá-la. Após essas mudanças, os testes indicaram um aumento significativo na precisão: a taxa de acerto em Matemática subiu para 95%, em História para 96% e em Música para 85%, demonstrando que a aplicação das técnicas de *prompting* contribuiu para uma melhora substancial na geração de respostas corretas, especialmente nas questões lógicas. Ainda que a repetição de perguntas geradas tenha se mantido como um desafio, o estudo conclui que a engenharia de *prompts* é essencial para o desenvolvimento de ferramentas educacionais baseadas em IA e destaca a importância de aperfeiçoamentos contínuos na interação entre usuários e modelos de linguagem. Os resultados evidenciam o potencial das técnicas de *prompting* em contextos educacionais, mas apontam a necessidade de armazenamento temporário das questões para evitar redundâncias e de uma abordagem progressiva para aprimorar ainda mais a efetividade do sistema.

Palavras-chave: Inteligência artificial; ChatGPT; Engenharia de *Prompts*; Educação.

ABSTRACT

This paper discussed the application of advanced prompt engineering techniques to optimize the automated creation of multiple-choice questions in the Creator4All software, developed by the company Multimídia Educacional. Aiming to improve the accuracy of responses generated by the widely-used ChatGPT language model, the study examines prompting strategies such as Zero-shot, Few-shot, Chain-of-Thought, and Chain-of-Verification. These techniques are employed to enhance the AI's comprehension and consistency in answering questions, particularly in complex tasks requiring logical reasoning. The research was conducted in two phases, involving the creation of 150 questions across the subjects of Mathematics, History, and Music, to assess the model's effectiveness and precision. In the first phase, using ChatGPT 3.5, the model demonstrated high accuracy in theoretical questions, achieving 93% correctness in History. However, its performance was more limited in areas requiring logical reasoning, with accuracy of 60% in Mathematics and 37% in Music, highlighting the need for adjustments. In the second phase, modifications were made to the prompts and software code, including the implementation of the Chain-of-Verification method, which verified the accuracy of the response before presenting it. After these changes, tests showed a significant improvement in precision: accuracy in Mathematics increased to 95%, in History to 96%, and in Music to 85%, demonstrating that the application of prompting techniques contributed to a substantial improvement in generating correct responses, particularly for logical questions. Although repeated questions remained a challenge, the study concludes that prompt engineering is essential for the development of AI-based educational tools and highlights the importance of continuous improvements in user-model interactions. The results emphasize the potential of prompting techniques in educational contexts but point to the need for temporary storage of questions to avoid redundancies and for a progressive approach to further enhance the system's effectiveness.

Keywords: Artificial Intelligence; ChatGPT; Prompt Engineering; Education.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
LLM	*Large Language Models*
IA	Inteligência Artificial
CoVe	*Chain-of-Verification*
W3C	*World Wide Web Consortium*
API	*Application Programming Interface*

SUMÁRIO

1. INTRODUÇÃO

A evolução das tecnologias de inteligência artificial tem transformado significativamente a maneira como interagimos com *softwares*. Um exemplo é o *software* Creator4All, que utiliza IA para gerar questões de múltipla escolha a partir de comandos textuais fornecidos pelos usuários. No entanto, a qualidade das respostas geradas depende diretamente da clareza e precisão dos comandos, ou *prompts*, fornecidos e das regras definidas anteriormente no código fonte. Nesse contexto, o presente estudo dedicou-se a investigar e aprimorar estratégias de escrita de *prompts*, visando alcançar maior consistência e taxa de acerto nos resultados produzidos pela ferramenta.

O trabalho buscou mapear metodologias de engenharia de *prompts* para identificar práticas eficazes que pudessem ser aplicadas no Creator4All. Além disso, foi realizada uma análise comparativa para avaliar os impactos das estratégias desenvolvidas, permitindo uma compreensão detalhada sobre como pequenas alterações na formulação dos *prompts* podem influenciar diretamente na qualidade das questões geradas. Os resultados dessa pesquisa têm potencial para contribuir significativamente na otimização do *software* e na experiência dos seus usuários, beneficiando diretamente o setor educacional.

2. MÉTODOS E TÉCNICAS DE ENGENHARIA DE *PROMPT*

2.1 *LARGE LANGUAGEM MODELS* (LLMS)

Nos últimos anos, a inteligência artificial (IA) e seus subprodutos têm ganhado grande destaque, com ampla repercussão, especialmente em relação ao ChatGPT, desenvolvido pela OpenAI. Essas IAs são conhecidas como *Large Language Models* (LLM), ou Modelos de Linguagem em Grande Escala, em tradução literal do inglês.

De acordo com Minaee et al., (2024, p. 1), as LLMs são modelos estatísticos de linguagem baseados em redes neurais pré-treinadas e em grande escala. O sucesso das LLMs é atribuído à sua capacidade de demonstrar um bom entendimento da linguagem, gerar habilidades, e, mais notavelmente, desenvolver habilidades emergentes, ou seja, capacidades adquiridas a partir de um contexto e de poucos exemplos como cita Kojima (2023, p. 1). Ainda segundo Kojima, as LLMs podem

resolver diversas tarefas ao serem condicionados com alguns exemplos, como na metodologia *few-shot*, ou com uma simples instrução descrevendo a tarefa, como na metodologia *zero-shot*. Esse processo de condicionamento é denominado *prompting*.

2.2 ENGENHARIA DE *PROMPTS*

Conforme Bsharat et al., (2024) no estudo *Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4*, o desenvolvimento de *prompts*, enquanto aspecto central da interação com Modelos de Linguagem em Grande Escala (LLMs), e a simplicidade proporcionada pela ausência da necessidade de ajuste fino dos modelos, evoluíram para um campo de estudo refinado. Esse avanço ressalta a complexa relação entre as entradas fornecidas pelos usuários e as respostas geradas pelos LLMs.

Estudos e interações realizadas com as LLMs identificaram que diferentes designs de prompts poderiam influenciar drasticamente o desempenho e os resultados dos modelos de linguagem, marcando o surgimento da engenharia de prompts.

A partir disso, houve uma expansão rápida do campo de estudo, onde é possível ver que com uma grande frequência surgem novas técnicas de prompts. Como destaque para essas novas técnicas podemos citar o *Zero-shot*, *Few-shot*, *Chain-of-Thought* e *Chain-Of-Verification*.

2.3 MÉTODOS DE APLICAÇÕES E BOAS PRATICAS

Com o crescente destaque que as inteligências artificiais têm alcançado nos últimos anos, a desenvolvedora da mais avançada IA disponível no mercado, o ChatGPT da OpenAI, disponibiliza uma plataforma intitulada "Cookbook", um repositório com dicas, orientações de uso e boas práticas.

De acordo com as recomendações do site, é fundamental que as instruções sejam claras, que tarefas complexas sejam divididas em subtarefas menores, que o modelo seja solicitado a explicar seus passos antes de fornecer uma resposta, que as respostas sejam justificadas e que múltiplas respostas sejam geradas, permitindo a seleção da melhor alternativa.

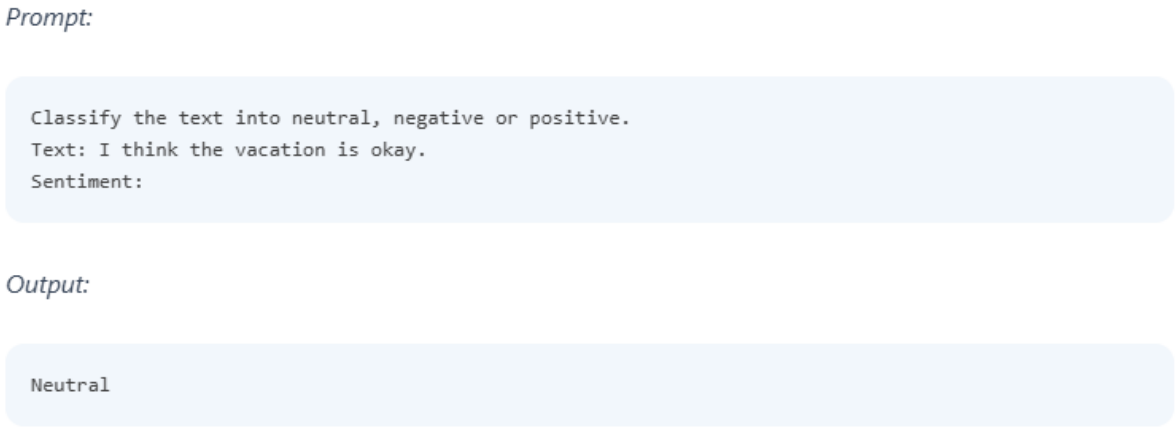
Com base nessas boas práticas indicadas pela desenvolvedora, a seguir serão apresentados alguns métodos que serão utilizados no desenvolvimento deste trabalho, bem como na formulação das regras aplicáveis ao modelo proposto.

2.3.1 *Zero-shot*

As atuais *Large Language Models* disponíveis no mercado foram treinadas com uma vasta quantidade de dados e demonstram a capacidade de seguir instruções de forma eficiente, permitindo a execução de diversas tarefas a partir de um único *prompt*.

O termo *zero-shot*, que pode ser traduzido como "sem tentativas anteriores ou único tiro", refere-se à obtenção de resultados com base em um único *prompt* inicial, sem que o modelo tenha sido previamente treinado ou recebido informações relacionadas.

Figura 1 – Exemplo do método *Zero-Shot*



Fonte: Zero-Shot Prompting | Prompt Engineering Guide (promptingguide.ai)

Observa-se que na Figura 1 não foi executado nenhum *prompt* de exemplo para a LLM e ainda sim ela conseguiu entender e retornar uma resposta que fez sentido.

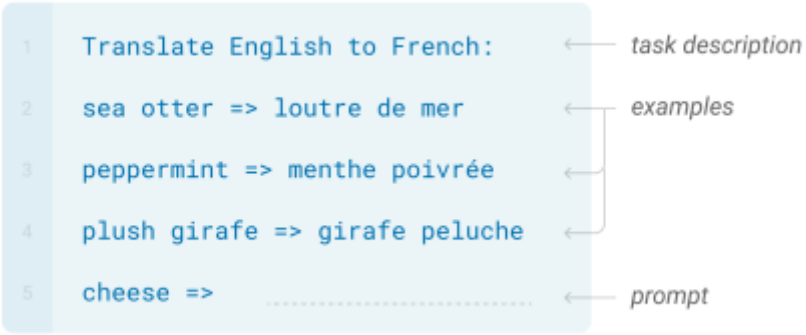
Embora esse método se destaque pela sua simplicidade de uso, ele apresenta limitações ao ser aplicado em tarefas mais complexas como menciona Wei et al, (2022) no artigo *Finetuned Language Model are Zero-shot Learners*. Em resposta a

essas dificuldades, pesquisadores ao redor do mundo têm desenvolvido uma variedade de outros métodos com o objetivo de aprimorar os resultados em tarefas que demandam maior complexidade.

2.3.2 *Few-shot* e *Chain-of-Thought*

Estas duas estratégias podem ser usadas separadamente, mas são comumente utilizadas em conjunto com outras técnicas pois podem aprimorar os resultados dos *prompts*: *Few-shot* e *Chain-of-Thought*. Embora as *Large Language Models* (LLMs) demonstrem uma performance satisfatória no método *zero-shot*, elas ainda enfrentam dificuldades em tarefas mais complexas, sobretudo em contas e raciocínio lógico. Como menciona Brown et al., (2020), para otimizar os resultados dos *prompts*, pode-se fornecer alguns exemplos ao modelo. Essa abordagem é denominada *few-shot*, ou "algumas tentativas", como é possível observar na figura 2, e permite que o modelo utilize os exemplos fornecidos para fundamentar suas respostas subsequentes. Pelo fato da natureza da técnica já demonstrar exemplos é possível classificar como *Chain-Of-Thought*, pois já vai criar uma corrente de pensamento.

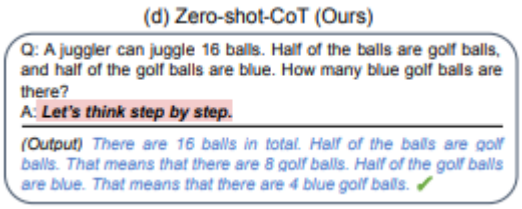
Figura 2 – Exemplo do método *Few-Shot*



Fonte: Brown et al., (2020)

Outro método relevante que tem ganhado popularidade recentemente é o *Zero-shot Chain-of-Thought* que envolve solicitar ao modelo que formule uma linha de raciocínio ou uma explicação antes de apresentar o resultado do *prompt*. A aplicação desse método, conforme demonstrado na figura 3 por Kojima et al., (2022), é a frase “Let’s think step by step”, que, em português, significa “Vamos pensar passo a passo”.

Figura 3 – Exemplo do método *Zero-Shot Chain-Of-Thought*



Fonte: Kojima et al, (2022).

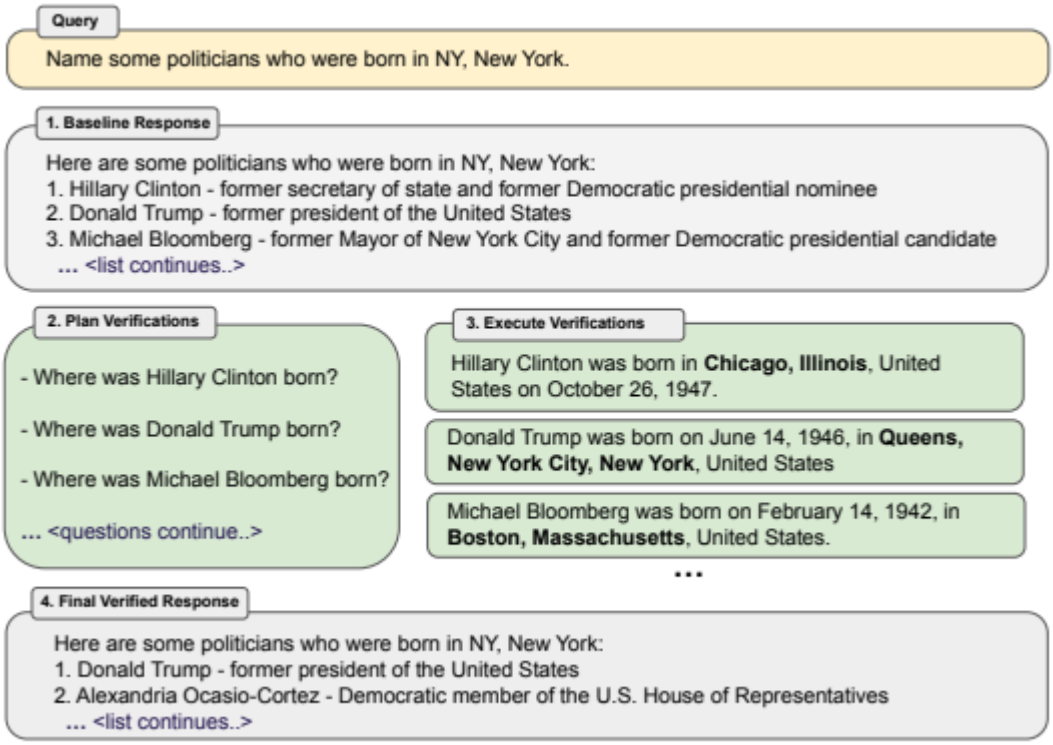
Essa abordagem incentiva o modelo a reconsiderar e reavaliar cada etapa da linha de raciocínio até chegar à resposta final. Na pesquisa de Kojima et al., (2022), foi utilizado um conjunto de dados de matemática, resultando em um aumento significativo na precisão dos resultados, que passou de 18% para 79%.

2.3.3 *Chain-of-Verification*

Outro método que pode ser utilizado para aprimorar as respostas geradas pelos *prompts* é a técnica conhecida como *Chain-of-Verification* (CoVe), ou Corrente de Verificação. Segundo Dhuliawala et al., (2023) em *Chain-of-Verification Reduces Hallucination in Large Language Models*, esse método é composto por quatro etapas: Resposta Base, Plano de Verificação, Execução de Verificações e Resposta Final.

A primeira etapa, denominada Resposta Base, refere-se à resposta inicial gerada a partir de um *prompt*. Após a obtenção dessa resposta, inicia-se a etapa do Plano de Verificação, que envolve a formulação de perguntas de verificação relacionadas à resposta inicial. Com as perguntas de verificação definidas, passa-se à etapa de Execução de Verificações, onde o modelo responde às perguntas, permitindo a identificação e validação de possíveis inconsistências em relação à resposta anterior. Por fim, chega-se à Resposta Final, que é gerada com base nos ajustes realizados após as verificações.

Figura 4 – Exemplo do método *Chain-Of-Verification*



Fonte: Dhuliawala et al., 2023, p.2.

Ainda de acordo com Dhuliawala et al. (2023), esse método reduz consideravelmente as alucinações que as IAs podem apresentar em suas respostas, embora sua implementação possa ser bastante trabalhosa e demorada.

2.4 CREATOR4ALL

O *Creator4all* é um *software* desenvolvido pela empresa Multimídia Educacional, a qual atua no segmento educacional há mais de 20 anos, com o objetivo de fornecer educação de qualidade integrada com tecnologia. Este *software* é voltado para a melhoria do processo de ensino-aprendizagem, incluindo a criação e o compartilhamento de materiais didáticos digitais inclusivos, bem como a gestão de desempenho em plataformas online por meio de relatórios.

O *software* disponibiliza uma ampla gama de funcionalidades que auxiliam os professores na produção e publicação de aulas, na gestão de tarefas, facilitando o acompanhamento do desempenho de alunos ou turmas, e na promoção da inclusão

digital, por meio de recursos de acessibilidade alinhados às diretrizes do *World Wide Web Consortium (W3C)*.

A funcionalidade que se destaca neste momento é o gerador de questões de múltipla escolha, disponível no *software* Creator4all Author, mas que requer uma licença especial para sua utilização. Inicialmente, a ferramenta já está funcional, embora necessite de alguns ajustes e refinamentos para alcançar melhores resultado. Atualmente, essa funcionalidade utiliza a API do ChatGPT 3.5 para a criação das questões a partir de um *prompt* e de parâmetros pré-configurados. Após esse processo, a ferramenta retorna a questão formulada juntamente com suas respectivas respostas.

3. METODOLOGIA

Na primeira fase deste estudo, foram geradas 150 questões distintas, distribuídas conforme segue: 50 questões de Matemática, 50 de História e 50 de Música. O objetivo desta fase foi avaliar a taxa de acerto do modelo ChatGPT tanto na geração das questões quanto na marcação das respostas corretas.

Nas três disciplinas foram escolhidos temas variados a fim de testar diferentes graus de complexidade na geração de questões e respostas.

Em contato com a equipe da Multimídia Educacional, foi disponibilizado um *token* de acesso à API do ChatGPT, bem como o código-fonte do *software* Creator4all Author para a realização dos testes de requisições. A versão da API do ChatGPT utilizadas será a gpt-3.5-turbo.

O ambiente de testes fornecido pela Multimídia Educacional consiste em um código PHP executado em um servidor local, sendo o servidor Apache, por meio do pacote XAMPP, com requisições realizadas utilizando a extensão *REST Client* no Visual Studio Code.

Na fase subsequente, denominada Fase 2, foram implementadas modificações no código do *software*, incluindo a adição de novas regras e alterações no modelo de retorno. Após essas alterações, foram novamente geradas 150 questões utilizando os mesmos temas selecionados, com o objetivo de verificar a assertividade do sistema.

Com os resultados de ambas as fases, foi possível avaliar se houve uma melhora na taxa de acerto do *software*.

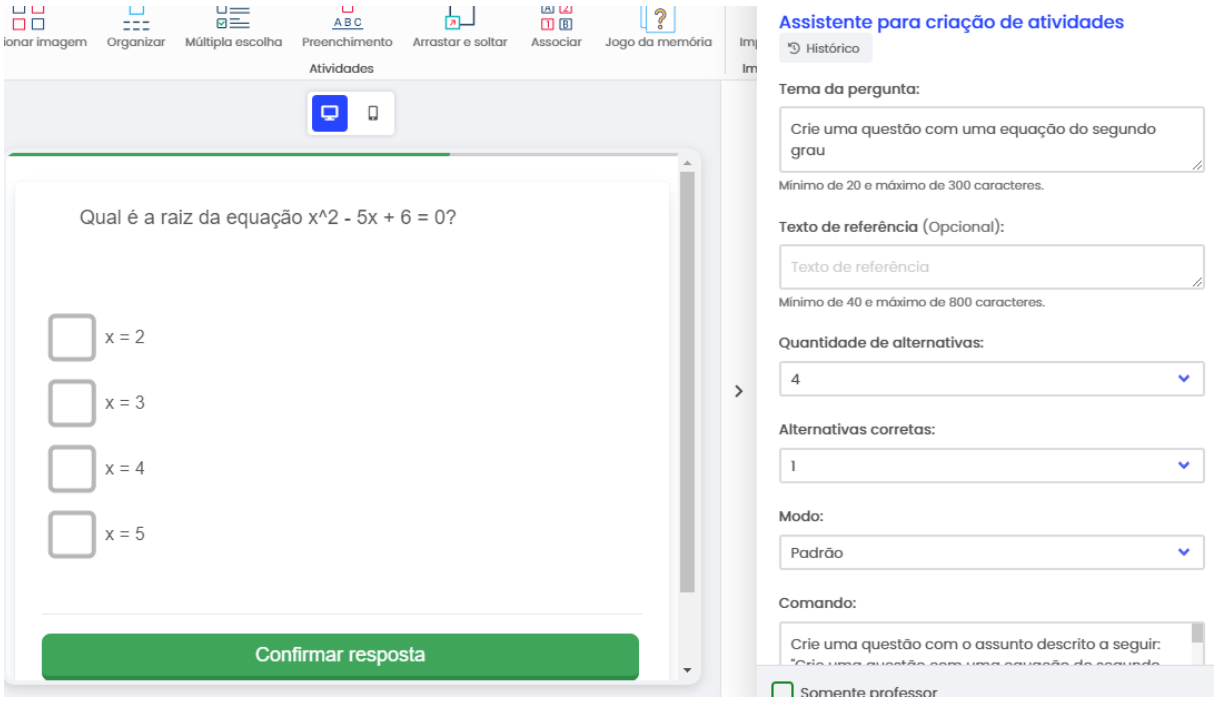
4. RESULTADOS E DISCUSSÕES

4.1 FASE 1

Nos primeiros testes realizados com o código e a *API* do ChatGPT 3.5, foi simulado o modelo atual utilizado pelo *software*. A requisição foi enviada por meio da extensão *REST Client*, e os resultados foram analisados. Consideraram-se corretas as perguntas geradas com quatro opções de resposta, sendo uma correta e três incorretas. Caso houvesse alguma repetição entre as respostas, a questão foi considerada incorreta. Também foi considerado como corretas questões repetidas que se enquadraram nos requisitos citados acima.

Observou-se que a *API* do ChatGPT compreendeu e gerou questões relacionadas aos *prompts* fornecidos, porém foram identificados alguns problemas. O primeiro problema, conforme ilustrado nas Figuras 5 e 6, foi a diminuição significativa da taxa de acerto em tarefas mais complexas, especialmente em questões que envolviam matemática, cálculos e lógica.

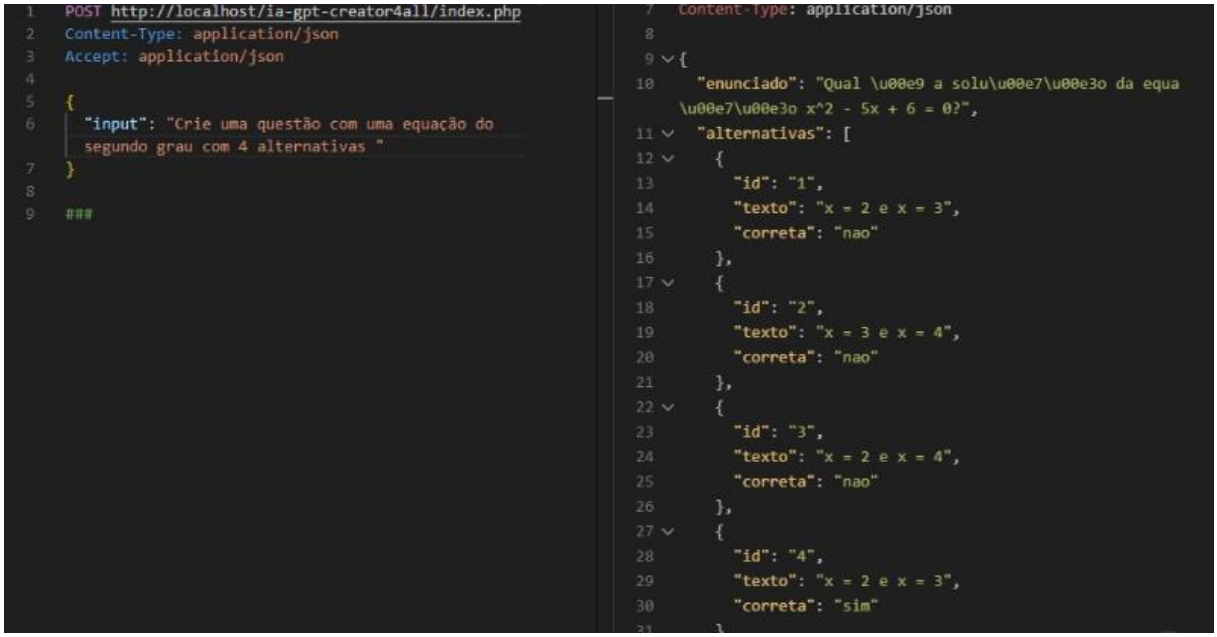
Figura 5 – Exemplo de pergunta gerada no *Creator4All*



Fonte: Elaboração própria.

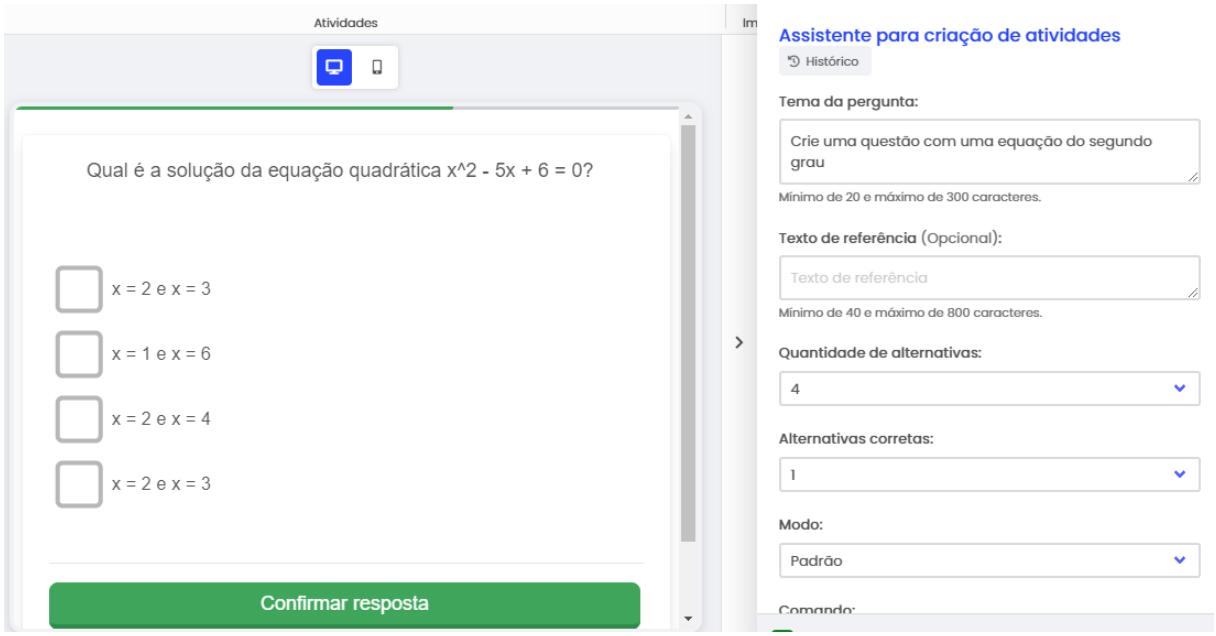
O segundo problema foi que a API considerou cada *prompt* inserido como um *zero-shot*, resultando, em muitas ocasiões, na geração de questões iguais às já apresentadas, com variações consideráveis nas opções de resposta e na correta, conforme as Figuras 6 e 7.

Figura 6 – Exemplo de pergunta gerado pelo código fonte



Fonte: Elaboração própria.

Figura 7 – Exemplo de pergunta gerada utilizando mesmo Prompt



Fonte: Autoria própria.

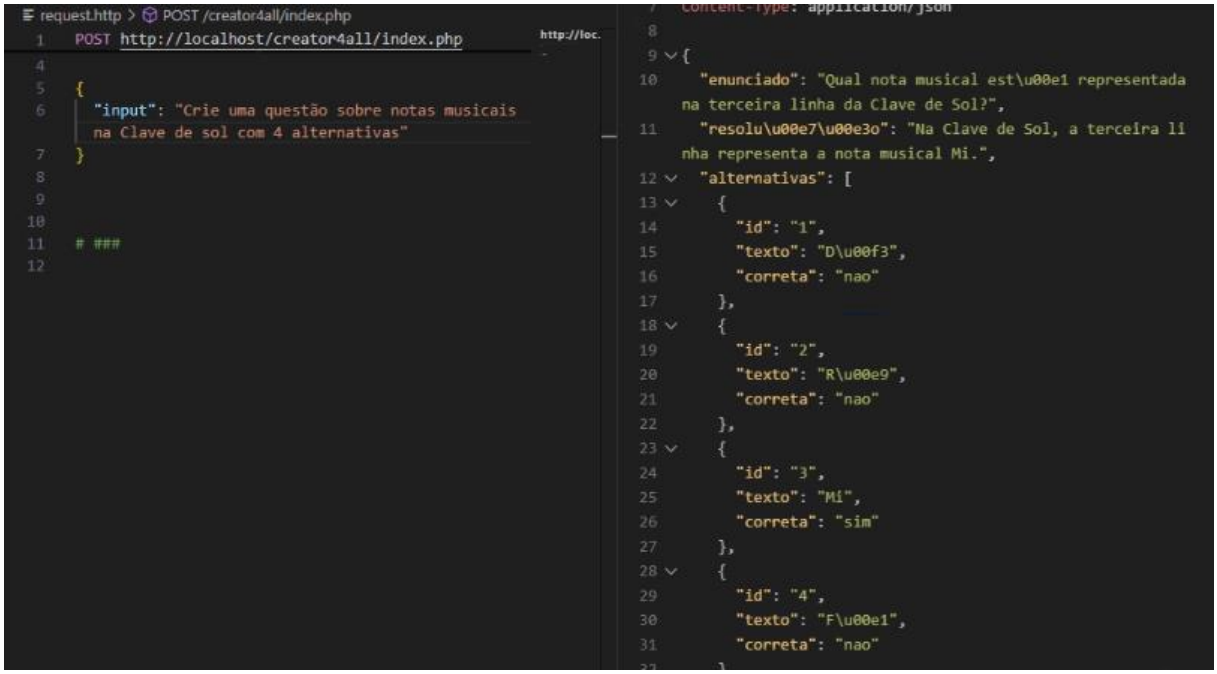
Na primeira fase, a taxa de acertos nas questões de matemática foi de 60%. Em questões mais teóricas, as perguntas e respostas foram bastante precisas. Em questões envolvendo conteúdos de pré-álgebra, álgebra e raciocínio lógico, o ChatGPT conseguiu gerar as perguntas, mas apresentou dificuldades ou erros nas respostas. Nos testes de história, obteve-se uma assertividade de 93%, demonstrando eficiência em questões teóricas e descritivas. Em música, a assertividade foi de 37%, concentrada principalmente em questões teóricas sobre a história da música, enquanto em questões de lógica envolvendo notas musicais o ChatGPT errou a maioria das respostas.

4.2 FASE 2

Após os testes iniciais, foram adicionadas algumas regras ao código-fonte para que o ChatGPT utilizasse o método *Zero-shot Chain-of-Thought* antes de retornar as questões e respostas do *prompt*. Foi reforçado nas regras do código fonte que apenas uma resposta por questão deveria estar correta e incluído um novo campo chamado ‘resolução’ no modelo *JSON* de retorno, garantindo que a resposta presente em ‘resolução’ estivesse entre as opções de resposta.

Na segunda fase de testes, após as modificações, a taxa de acerto foi de 95% em questões de matemática, com melhora significativa na geração de respostas corretas, especialmente em álgebra. Em história, a taxa de acerto foi de 96%, com um aumento de 3% em relação à fase anterior. Apesar dos bons resultados, algumas respostas da API em determinados contextos apresentaram estranheza, embora não fossem completamente incorretas. Em música, a taxa de acerto aumentou para 85%, demonstrando melhora considerável. Questões mais teóricas tiveram bons resultados, mas o modelo continuou a demonstrar dificuldade em questões de lógica, conforme a Figura 8.

Figura 8 – Exemplo da dificuldade em questões lógicas.



Fonte: Elaboração própria.

Embora tenha havido melhora considerável nos resultados dos testes, o problema de geração de perguntas repetidas ainda persiste. Para resolver esse problema, seria necessária a criação de um arquivo temporário para armazenar as questões geradas e realizar uma verificação antes de cada nova geração de questão.

.

5. CONSIDERAÇÕES FINAIS OU CONCLUSÃO

Nos testes realizados, é importante relatar que o estudo apresenta algumas limitações. O foco foi direcionado para os diferentes resultados possíveis ao realizar modificações nos *prompts* e configurações, contudo, a quantidade de testes e questões foi baixa, totalizando 300 testes. Com uma base maior, diferentes resultados podem ser observados.

O ChatGPT é uma referência entre as inteligências artificiais e, apesar das limitações do trabalho, demonstrou nos testes e no *software* utilizado facilidade com perguntas e respostas teóricas. No entanto, ao realizar testes que envolvem raciocínio lógico, apresentou inconsistências nos resultados. Com a adição de campos e arquivos que o ChatGPT pudesse consultar ou verificar antes de retornar uma resposta, utilizando um modelo de *Zero-shot Chain-of-Thought*, o desempenho em perguntas e respostas que envolvem raciocínio lógico melhorou consideravelmente reforçando e concordando com os resultados obtidos por Kojima et al., (2022).

Para os casos de repetição, observou-se uma melhora na consistência das respostas geradas. Contudo, ainda há ocorrências de perguntas idênticas, sendo necessário mencionar explicitamente no *prompt* a questão anterior para evitar repetições. Visando aprimorar o funcionamento, uma solução potencial seria a implementação de um sistema de armazenamento que registre as perguntas já geradas. Esse arquivo seria consultado durante cada novo processo de geração, permitindo verificar a existência de questões duplicadas e, assim, prevenir sua repetição.

Outro método testado foi o *Few-Shot*, que demonstrou limitações devido à restrição de *tokens* por requisição e resposta. Contudo, considera-se viável a implementação de um campo adicional onde seja possível inserir um exemplo de exercício, permitindo que o modelo, como o ChatGPT, desenvolva outros exercícios com base nesse exemplo.

Apesar das melhorias observadas nas gerações e escolhas de perguntas e respostas, constata-se que o modelo geral ainda apresenta margem para aprimoramento, especialmente em temas que demandam raciocínio lógico. Nesse contexto, destacam-se possíveis caminhos de implementação e melhora que visem seu uso para fins educacionais, com o objetivo de promover inovação, estimular o

aprendizado dos alunos e oferecer suporte ao educador, facilitando e enriquecendo o processo de ensino-aprendizado.

REFERÊNCIAS

BENTO, A. **Como fazer uma revisão da literatura: considerações teóricas e práticas.** *Revista JA (Associação Académica da Universidade da Madeira)*, n. 65, ano VII, p. 42-44, maio 2012. ISSN 1647-8975.

BROWN, Tom B.; MANN, Benjamin; RYDER, Nick; SUBBIAH, Melanie; KAPLAN, Jared; DHARIWAL, Prafulla; NEELAKANTAN, Arvind; SHYAM, Pranav; SASTRY, Girish; ASKELL, Amanda; AGARWAL, Sandhini; HERBERT-VOSS, Ariel; KRUEGER, Gretchen; HENIGHAN, Tom; CHILD, Rewon; RAMESH, Aditya; ZIEGLER, Daniel M.; WU, Jeffrey; WINTER, Clemens; HESSE, Christopher; CHEN, Mark; SIGLER, Eric; LITWIN, Mateusz; GRAY, Scott; CHESS, Benjamin; CLARK, Jack; BERNER, Christopher; MCCANDLISH, Sam; RADFORD, Alec; SUTSKEVER, Ilya; AMODEI, Dario. **Language Models are Few-Shot Learners**. 2020. Disponível em: https://arxiv.org/abs/2005.14165. Acesso em: 10 nov. 2024.

BSHARAT, Sondos Mahmoud; MYRZAKHAN, Aidar; SHEN, Zhiqiang. **Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4.** Disponível em: https://github.com/VILA-Lab/ATLAS. Acesso em: 10 nov. 2024.

CREATOR4ALL. **Creator4All Desktop: múltipla escolha e IA.** Disponível em: https://wiki.creator4all.com/guia/creator4all-desktop/multipla-escolha-ia.html. Acesso em: 10 mai. 2024.

CREATOR4ALL. **Creator4All Desktop: requisitos mínimos.** Disponível em: https://wiki.creator4all.com/guia/creator4all-desktop/requisitos-minimos.html. Acesso em: 10 mai. 2024.

CREATOR4ALL. **Wiki Creator4All.** Disponível em: https://wiki.creator4all.com/. Acesso em: 10 mai. 2024.

DHULIAWALA, Shehzaad; KOMEILI, Mojtaba; XU, Jing; RAILEANU, Roberta; LI, Xian; CELIKYILMAZ, Asli; WESTON, Jason. **Chain-of-Verification Reduces Hallucination in Large Language Models.** 2023. Disponível em: https://arxiv.org/abs/2309.11495. Acesso em: 10 nov. 2024.

KOJIMA, Takeshi; GU, Shixiang Shane; REID, Machel; MATSUO, Yutaka; IWASAWA, Yusuke. **Large Language Models are Zero-Shot Reasoners**. 2022. Disponível em: https://arxiv.org/abs/2205.11916. Acesso em: 10 nov. 2024.

MINAEE, Shervin; MIKOLOV, Tomas; NIKZAD, Narjes; CHENAGHLU, Meysam; SOCHER, Richard; AMATRIAIN, Xavier; GAO, Jianfeng. **Large Language Models: A Survey.** 2024. Disponível em: https://arxiv.org/abs/2402.06196. Acesso em: 10 nov. 2024.

MULTIMÍDIA EDUCACIONAL. **Multimídia Educacional.** Disponível em: https://www.multimidiaeducacional.com.br/. Acesso em: 10 nov. 2024.

OPENAI. **Strategy: Use external tools.** 2023. Disponível em: https://platform.openai.com/docs/guides/prompt-engineering/strategy-use-external-tools. Acesso em: 10 nov. 2024.

OPENAI. **Techniques to improve reliability.** 2023. Disponível em: https://cookbook.openai.com/articles/techniques_to_improve_reliability. Acesso em: 10 nov. 2024.

WEI, Jason; BOSMA, Maarten; ZHAO, Vincent Y.; GUU, Kelvin; YU, Adams Wei; LESTER, Brian; DU, Nan; DAI, Andrew M.; LE, Quoc V. Finetuned **Language Models Are Zero-Shot Learners.** 2021. Disponível em: https://arxiv.org/abs/2109.01652. Acesso em: 10 nov. 2024.