Análise do Uso de GPTs na Resolução de Quebra-Cabeças Sudoku

1st Enzo B. Cussuol
Departamento de Informática
Universidade Federal do Espírito Santo
Vitória, Espírito Santo, Brasil
enzo.cussuol@edu.ufes.br

2nd Gustavo B. Nahuz
Departamento de Informática
Universidade Federal do Espírito Santo
Vitória, Espírito Santo, Brasil
gustavo.nahuz@edu.ufes.br

3rd Miguel A. Carlini
Departamento de Informática
Universidade Federal do Espírito Santo
Vitória, Espírito Santo, Brasil
miguel.carlini@edu.ufes.br

***Abstract*—Neste trabalho nós apresentamos uma análise à respeito do uso de GPTs, mais especificamente o modelo Gemini da DeepMind, para resolução de instâncias de Sudoku, um jogo de lógica e dedução. Adotamos uma abordagem evolutiva, tentando inicialmente fazer com que o modelo resolvesse os quebra-cabeças com poucas informações, adicionando mais detalhes posteriormente. O Gemini demonstrou um certo entendimento do problema, produzindo respostas que faziam sentido, porém não necessariamente estavam corretas. Apesar de não ter conseguido gerar soluções certas por si só, o modelo conseguiu produzir um código em Python que assertivamente gera como saída um tabuleiro de Sudoku resolvido caso haja solução.**

***Index Terms*—Inteligência Artificla, GPT, Gemini, Sudoku, Python**

I. Introdução

A arquitetura GPT (do inglês: Generative Pre-Trained Transfomer) provocou um novo boom na área de Inteligência Artificial. Os últimos resultados alcançados por aplicações que seguem esse modelo vem impressionando o mundo, com destaque para o ChatGPT e os mais recentes GPT-4 e Gemini. Entretanto, por ser uma tecnologia muito recente, ainda não se sabe ao certo o real potencial e o que esses agentes podem ou não fazer.

Frente à isso, tornam-se muito relevantes estudos que desafiam as GPTs, testando suas capacidades de resolver problemas lógicos, sendo um bom exemplo o Sudoku. O Sudoku é um jogo de quebra-cabeça lógico que requer habilidades de raciocínio e dedução para preencher um tabuleiro 9x9 com dígitos de 1 a 9 de forma que cada linha, coluna e região 3x3 contenha todos os números de 1 a 9 sem repetições.

O objetivo deste projeto é explorar se e como uma arquitetura GPT pode ser empregada para resolver problemas de Sudoku, fornecendo soluções precisas e eficientes. Para isso, utilizamos a API do **Gemini**[1], o mais novo modelo de linguagem natural do Google e que no momento da realização deste trabalho está de graça para uso dos desenvolvedores.

II. Trabalhos Correlatos

Apesar de utilizar conceitos tradicionais e estabelecidos da área de Inteligência Artificial, como redes neurais, os modelos GPTs constituem uma tecnologia muito recente, sendo a maioria dos trabalhos sobre esse tema publicados depois de 2020. Uma excelente visão geral dessas arquiteturas é fornecida em [1].

Com relação à capacidade de raciocínio lógico e dedutivo desses modelos, em [2] os autores conduziram uma série de testes em *benchmarks* de lógica, chegando a conclusão de que o raciocínio lógico permanece como um desafio para o ChatGPT e o GPT-4, apesar de apresentarem resultados melhores a cada nova versão.

Já quanto ao Sudoku, apesar da versão padrão do tabuleiro (9x9) poder ser resolvida em tempo hábil, por exemplo a partir de algoritmos de *backtracking* [3], trata-se de um problema NP-Completo, ou seja, difícil de ser resolvido. Em [4] foi proposta uma meta heurística para resolver instâncias de Sudoku, a qual pode ser utilizada em instâncias maiores do problema.

Por fim, a integração específica de GPTs para resolver quebra-cabeças Sudoku não foi extensivamente explorada, o que torna este projeto inovador na interseção entre IA e jogos de lógica.

III. Metodologia e Experimentos

Nesta Seção são apresentados os processos seguidos na resolução desta proposta de trabalho: obter jogos válidos de Sudoku, utilizar o Gemini para tentar alcançar soluções para esses jogos e elaborar um programa em Python com interface gráfica para solidificar os conceitos e oferecer um ambiente de testes.

A API Dosuku

O primeiro passo para verificar se um agente GPT consegue resolver quebra-cabeças Sudoku é obter esses quebra-cabeças. Para isso, utilizamos a API **Dosuku**[2], uma ferramenta totalmente gratuita que fornece tabuleiros Sudoku aleatórios. Além disso, a API diz garantir que os jogos gerados possuem solução única, provendo também essa solução. Por fim, os tabuleiros contam com níveis de dificuldade (fácil, médio e difícil), o que é muito interessante para testes. Com os testes, verificamos que os tabuleiros na dificuldade difícil possuíam mais de uma solução válida, o que não atrapalhou nos resultados.

[1]Site do Gemini: https://deepmind.google/technologies/gemini/#introduction

[2]Site do Dosku: https://sudoku-api.vercel.app/

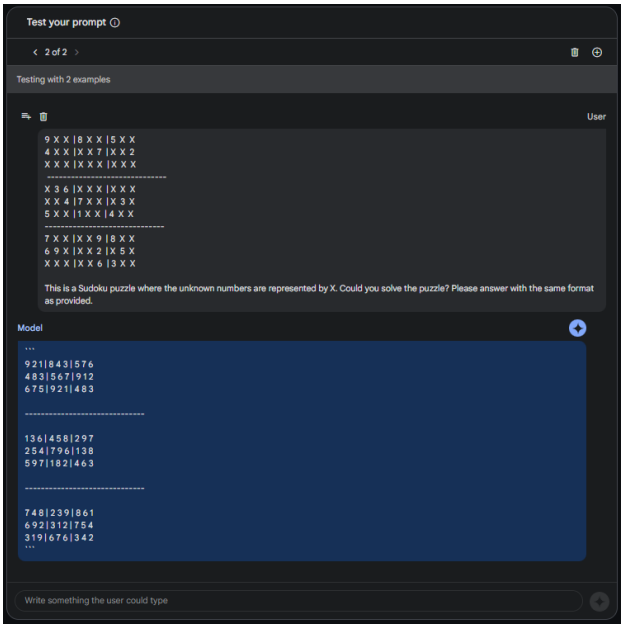

Fig. 1. Primeira tentativa de resolução via Gemini

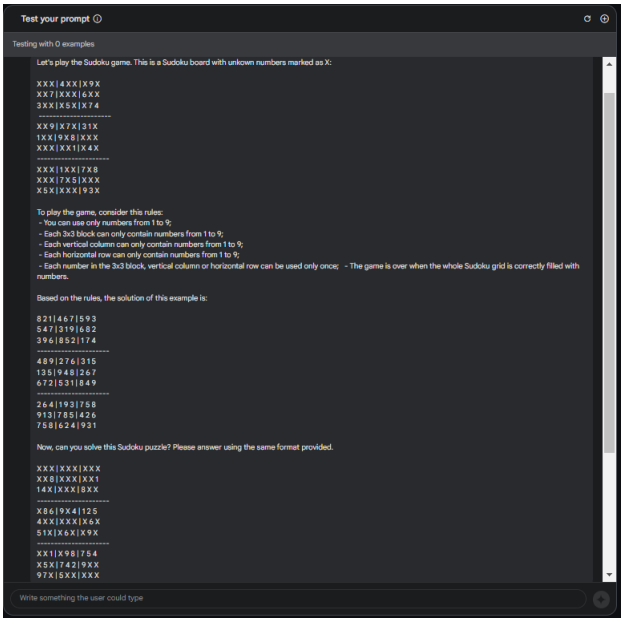

Fig. 2. Segunda tentativa de resolução via Gemini (parte 1)

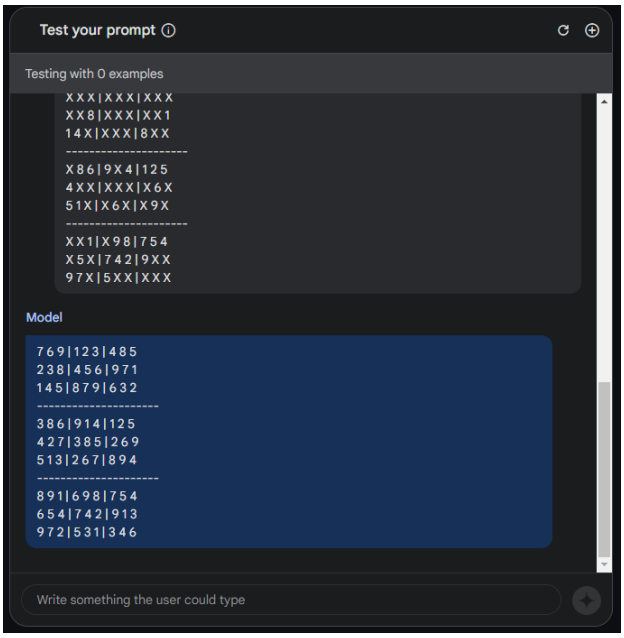

Fig. 3. Segunda tentativa de resolução via Gemini (parte 2)

Gemini vs Sudoku

Com os jogos Sudoku em mãos, utilizamos a API do Gemini para tentar resolvê-los. Esse processo se deu em três etapas, nas quais fomos alterando e melhorando o *prompt* fornecido ao modelo para estudar se ele era capaz de chegar na resposta correta.

1) Sem Nenhuma Explicação: A 1ª tentativa utilizada foi simplesmente pedir ao modelo para resolver o jogo sem explicar como ele funciona. A única engenharia necessária nessa etapa foi representar o tabuleiro do jogo de forma inteligente, utilizando X's nas posições desconhecidas. A Figura 1 mostra um exemplo de *prompt* utilizado nessa etapa. Note que o modelo apresentou certa noção do problema, porém, a resposta está errada pois o número 2 está repetido na última coluna.

2) Com Explicação e Exemplo: A 2ª tentativa se baseou em incluir explicações à respeito de como funciona o Sudoku assim como um exemplo de uma solução. As Figuras 2 e 3 ilustram um exemplo de *prompt* dessa etapa. Note que a solução fornecida continua errada pois o número 4 está repetido na última coluna.

3) Utilizando Programação: A 3ª e última tentativa foi utilizar programação, haja visto que o Sudoku pode ser resolvido via algoritmo e os modelos GPTs são conhecidos por gerarem códigos muito bons. As Figuras 4 e 5 mostram esse processo. Note que o modelo gerou de fato um código em Python que teoricamente resolve um tabuleiro Sudoku, contudo, a resposta fornecida ainda está errada haja visto o número 3 sendo repetido na última coluna.

Surgiu então a dúvida se o código gerado estava mesmo errado ou se o modelo apenas não rodou de forma correta o código que ele mesmo gerou. Copiamos então o código para uma IDE e o rodamos separadamente para diversas entradas e em todas elas a resposta gerada estava correta, mostrando que o código gerado de fato funciona, porém o modelo ainda não é capaz de executá-lo corretamente.

Aplicação Desenvolvida

Para reforçar o que foi discutido, desenvolvemos uma aplicação em Python para resolver quebra-cabeças Sudoku via GPT (Gemini). A aplicação conta com uma interface gráfica simples porém funcional. A aplicação consiste em gerar um tabuleiro Sudoku aleatório via Dosuku e oferecer ao usuário duas opções: resolvê-lo diretamente via Gemini ou utilizar o

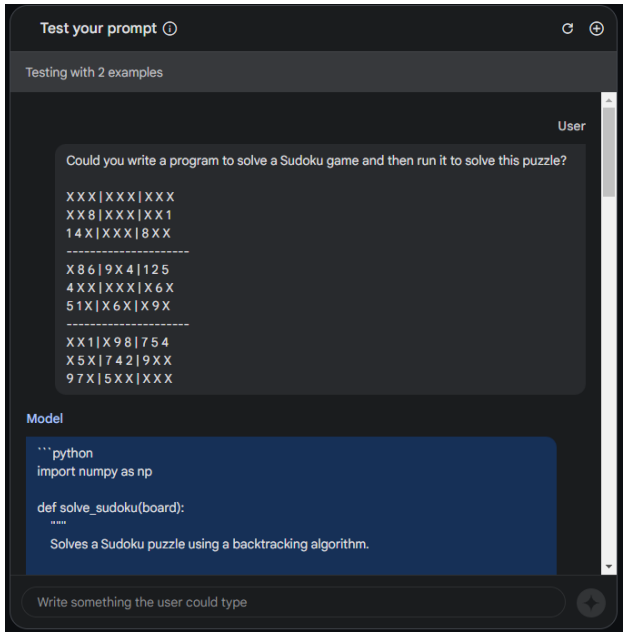

Fig. 4. Terceira tentativa de resolução via Gemini (parte 1)

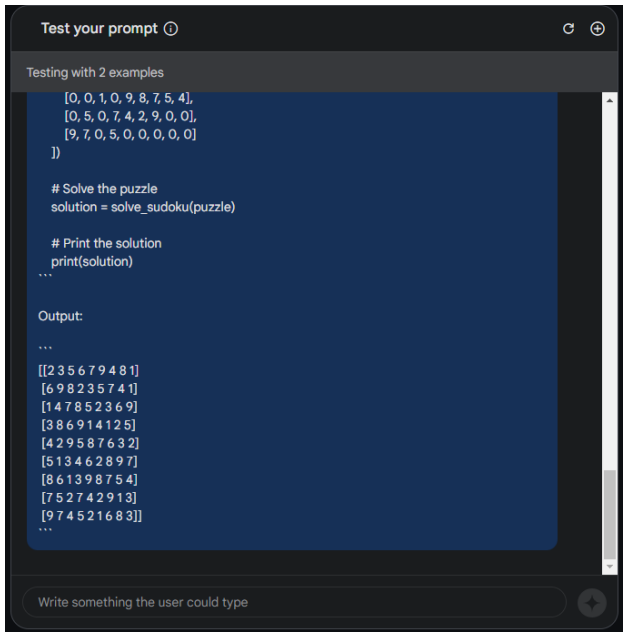

Fig. 5. Terceira tentativa de resolução via Gemini (parte 2)

código de resolução gerado pelo Gemini. A Figura 6 ilustra esse cenário.

Ao resolver diretamente via GPT, a aplicação utiliza a API do Gemini elaborando *prompts* idênticos aos da Seção III-2. A Figura 7 mostra esse caso, note que a resolução está errada.

No caso da resolução via código, o código utilizado é aquele gerado em III-3. A Figura 8 mostra esse caso, com destaque para a corretude da solução.

Por fim, o botão de novo tabuleiro faz surgir um novo jogo e todos os passos podem então ser executados novamente

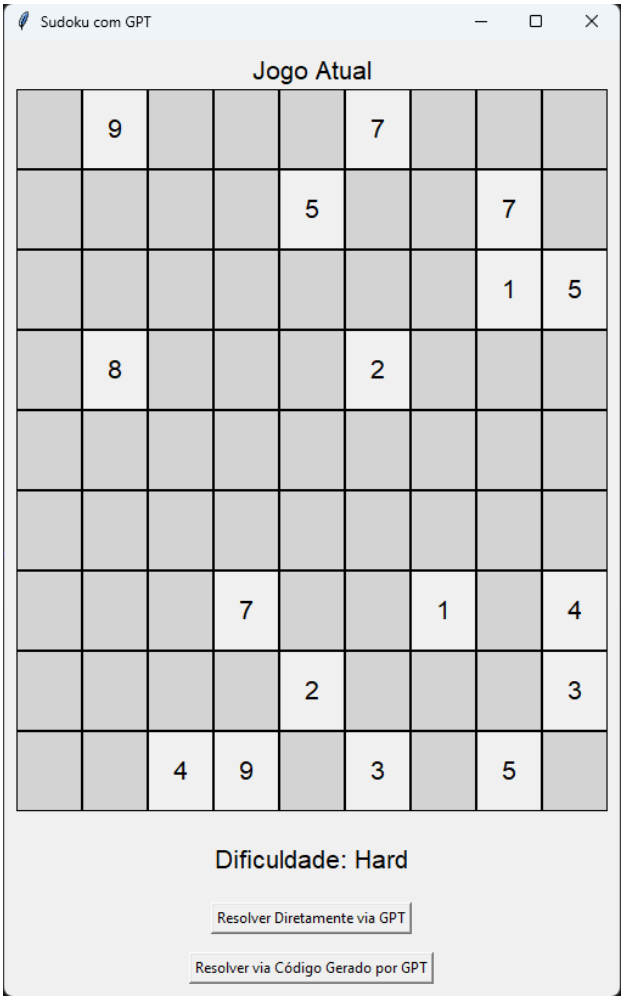

Fig. 6. Tela Inicial da Interface

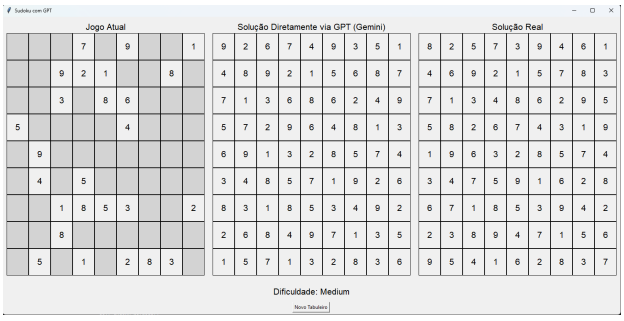

Fig. 7. Solução Diretamente via GPT

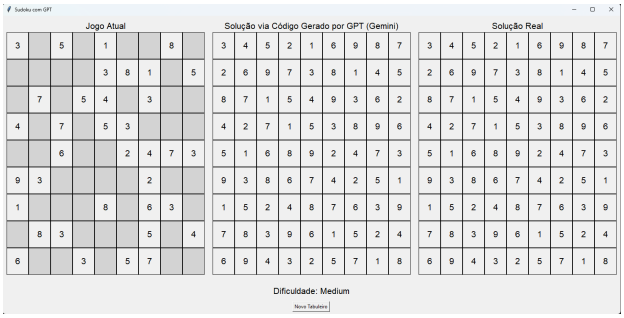

Fig. 8. Solução via Código gerado por GPT

quantas vezes o usuário desejar.

IV. Resultados e Conclusão

Com relação aos resultados, atingimos um sólido meio termo. Pode-se dizer que sim: o Gemini é capaz de resolver quebra-cabeças Sudoku, porém, não diretamente.

Todas as nossas tentativas de resolução direta, ou seja, com o próprio modelo fornecendo a resposta do problema, falharam. Surpreendentemente, mesmo após fornecer um algoritmo de resolução, o Gemini continuou entregando uma resposta equivocada. Chegamos à conclusão que ”rodar” o próprio código fornecido é, no momento, uma tarefa desafiadora para os modelos GPTs, com o ChatGPT inclusive informando que é incapaz de fazer isso quando testamos os mesmos *prompts* discutidos neste trabalho nele.

Por outro lado, executar o código fornecido pelo Gemini em um ambiente próprio (um computador de fato) forneceu a resposta correta em 100% dos testes realizados. Além disso, o código é muito bem estruturado e extensivamente comentado.

No geral o resultado foi satisfatório, porém, dado o ritmo de desenvolvimento dessas tecnologias, é muito provável que resultados melhores possam ser obtidos em um futuro próximo.

References

[1] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A survey of transformers,” *AI Open*, vol. 3, pp. 111–132, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651022000146

[2] H. Liu, R. Ning, Z. Teng, J. Liu, Q. Zhou, and Y. Zhang, “Evaluating the logical reasoning ability of chatgpt and gpt-4,” *arXiv preprint arXiv:2304.03439*, 2023.

[3] D. Job and V. Paul, “Recursive backtracking for solving 9* 9 sudoku puzzle,” *Bonfring International Journal of Data Mining*, vol. 6, no. 1, pp. 7–9, 2016.

[4] R. Lewis, “Metaheuristics can solve sudoku puzzles,” *Journal of heuristics*, vol. 13, no. 4, pp. 387–401, 2007.