



# Análise de dados em blockchain com modelos de linguagem generativa

*O. Anovazzi      L. E. Oliveira      A. M. Souza*

Relatório Técnico - IC-PFG-24-32

Projeto Final de Graduação

2024 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

# Análise de dados em blockchain com modelos de linguagem generativa

Otávio Anovazzi      Lucas Eduardo Ramos de Oliveira      Allan M. de Souza

## Resumo

Este trabalho visa democratizar o acesso a informações públicas e técnicas sobre blockchains, utilizando técnicas de análise de dados com modelos de linguagem generativa. A proposta inclui o uso de visão computacional e modelos de linguagem multimodais para explorar dados extraídos da rede blockchain através de APIs, permitindo uma interpretação mais acessível e detalhada dessas informações. A realização do estudo baseou-se no projeto open-source *mempool* [1], disponível no GitHub, que fornece dados essenciais para a implementação das análises e desenvolvimento de soluções inovadoras para consultas em blockchain.

Com base em diversos testes, foi realizado um comparativo entre várias configurações para análise de dados de blockchain e geração de texto, com o intuito de identificar a abordagem mais eficiente. Observou-se que a configuração com melhor desempenho foi o modelo GPT-4-Vision-Preview integrado com web search e temperatura 0.8, destacando-se pela precisão e rapidez na análise dos dados. A diferença de performance entre essa configuração e a menos eficiente alcançou aproximadamente 24%, de acordo com parâmetros de avaliação como pontuação de extensão (*length\_score*), semelhança com linguagem humana (*human\_likeness*), relevância (*relevance*) e precisão factual (*factual\_accuracy*). Esses resultados demonstram a importância de escolher parâmetros e modelos adequados para otimizar a interpretação de dados em redes blockchain.

## 1 Introdução

O crescimento exponencial das tecnologias de blockchain tem trazido consigo uma abundância de dados disponíveis em redes distribuídas. Cada transação, contrato inteligente ou bloco criado gera informações que, embora estejam publicamente acessíveis, frequentemente se apresentam em formatos complexos e pouco intuitivos. Esse cenário cria uma lacuna entre a vastidão dos dados disponíveis e a capacidade de compreendê-los, especialmente por parte de públicos não especializados. Consequentemente, a democratização do acesso a dados em blockchain, com uma apresentação clara e de fácil entendimento, torna-se cada vez mais relevante.

A análise de dados no contexto de blockchain enfrenta desafios específicos, tais como a necessidade de processar grandes volumes de informações em tempo real e a diversidade de tipos de dados envolvidos, que vão desde transações financeiras até metadados de redes distribuídas. Para abordar esses desafios, os avanços em modelos de linguagem generativa e multimodais oferecem uma oportunidade inovadora. Modelos como GPT e similares,

originalmente desenvolvidos para processamento de linguagem natural, têm se mostrado promissores também para a análise e síntese de dados complexos em diferentes domínios, inclusive o de blockchain. Além disso, a aplicação de visão computacional e de modelos multimodais – que integram dados textuais e visuais – expande a capacidade de interpretar e traduzir dados de maneira compreensível e acessível.

Este trabalho propõe uma abordagem para analisar dados de blockchain por meio de modelos de linguagem generativa, apoiados por técnicas de visão computacional e sistemas multimodais. Para isso, utiliza-se o projeto open-source *mempool* [1], que fornece dados da rede e permite a exploração detalhada e prática de informações públicas de blockchain. Através de uma série de experimentos, este estudo busca identificar as configurações que melhor balanceiam desempenho e precisão na geração de informações interpretáveis.

A pesquisa também contempla uma análise comparativa entre diferentes configurações de modelos de linguagem e parâmetros de análise de dados, visando determinar a configuração mais eficaz para o contexto específico de blockchain.

## 2 Justificativa

A justificativa para este trabalho reside na importância crescente de interpretar dados de blockchain de forma eficiente e acessível. À medida que o uso de redes descentralizadas e contratos inteligentes aumenta, os dados gerados por essas tecnologias também se multiplicam, exigindo ferramentas capazes de extrair e sintetizar informações complexas de maneira acessível para públicos amplos. A maioria dos dados em blockchain, embora esteja disponível publicamente, é armazenada em formatos técnicos que dificultam a compreensão, especialmente para usuários não técnicos. Essa complexidade representa uma barreira significativa para a democratização do acesso à informação e limita o potencial de inovação colaborativa e de transparência que o blockchain promete.

A democratização do acesso a dados de blockchain, portanto, é essencial para popularizar ainda mais a tecnologia e torná-la acessível a um público diverso. Ferramentas que traduzam dados complexos de blockchain em informações claras e compreensíveis são fundamentais para que a tecnologia seja verdadeiramente inclusiva. Nesse sentido, os avanços em modelos de linguagem generativa, como os modelos da família GPT, oferecem uma solução inovadora. Esses modelos, projetados inicialmente para processamento de linguagem natural, têm demonstrado eficácia na síntese e interpretação de grandes volumes de dados, o que os torna particularmente adequados para enfrentar os desafios do blockchain.

A escolha do projeto open-source *mempool* [1] como fonte de dados para esta análise reforça ainda mais a relevância deste estudo, já que essa plataforma fornece dados atualizados e em tempo real da rede blockchain, essenciais para entender as transações e fluxos da rede. Combinando essas informações com modelos de linguagem e sistemas multimodais, espera-se criar um método eficaz para transformar dados técnicos em uma representação mais acessível e interpretável.

Este estudo se justifica, portanto, pela necessidade de tornar a tecnologia blockchain mais acessível e compreensível para um público não especializado. Ao explorar o potencial de modelos de linguagem generativa para traduzir dados técnicos em insights acessíveis,

esta pesquisa visa contribuir para a popularização e democratização do acesso ao blockchain, permitindo que mais pessoas e organizações aproveitem os benefícios dessa tecnologia emergente.

## **3 Objetivos**

### **3.1 Extração de dados transparentes e interpretáveis da rede blockchain**

A extração de dados de redes blockchain deve ser clara e acessível, especialmente devido à complexidade técnica e ao volume de informações gerados continuamente. Este objetivo visa desenvolver um método que possibilite a coleta e organização de dados de maneira a preservar sua integridade, ao mesmo tempo em que facilita sua interpretação para usuários com diferentes níveis de conhecimento técnico. Esse método precisa garantir que os dados sejam extraídos e estruturados de forma que permitam a visualização e análise de forma simples e compreensível.

Além disso, o método deve promover a transparência no processo de extração, fornecendo aos usuários uma visão detalhada de cada etapa, desde a coleta inicial até a apresentação dos dados. Por conta da transparência todas as ferramentas utilizadas para extração de dados são construídas utilizando código aberto.

### **3.2 Conversão dados numéricos e gráficos complexos em descrições em linguagem natural**

A conversão de dados numéricos e gráficos em linguagem natural visa facilitar a compreensão de informações complexas, permitindo que insights relevantes sejam acessíveis para um público mais amplo. Este objetivo envolve o desenvolvimento de uma abordagem capaz de traduzir dados técnicos – como valores numéricos, tendências e visualizações gráficas – em descrições textuais claras e detalhadas. Isso elimina a necessidade de conhecimentos especializados para interpretar os dados, tornando a análise de blockchain mais acessível.

Além de simplificar a compreensão, essa conversão promove maior inclusão e democratização dos dados, pois informações importantes sobre blockchain podem ser transmitidas em linguagem natural, reduzindo barreiras de acesso e engajamento. Com descrições textuais, espera-se que o sistema ajude diferentes tipos de usuários a acompanhar e interpretar dados, aprimorando sua experiência e compreensão em relação ao funcionamento da rede blockchain.

### **3.3 Análise comparativa entre modelos e parâmetros de geração de linguagem**

A análise comparativa envolve testar múltiplos modelos e parâmetros de geração de linguagem, investigando como cada um processa e traduz dados técnicos em descrições textuais interpretáveis. Essa abordagem permitirá observar variações na qualidade e clareza dos resultados, determinando as configurações mais eficazes para o contexto específico de blockchain.

Além disso, o estudo comparativo ajudará a estabelecer padrões de desempenho para a aplicação de modelos de linguagem na análise de dados distribuídos. Ao entender as vantagens e limitações de cada configuração, será possível otimizar o uso de modelos de linguagem em análises futuras, contribuindo para uma maior precisão e acessibilidade dos dados de blockchain apresentados em linguagem natural.

### 3.4 Desenvolver uma ferramenta gratuita e de fácil uso para análise de dados em blockchain

A ferramenta deve oferecer uma interface amigável, que simplifique o processo de interpretação de dados e apresente informações complexas de forma clara e organizada, permitindo que qualquer pessoa possa explorar e compreender os dados disponíveis na rede blockchain.

## 4 Desenvolvimento do Trabalho

### 4.1 Pesquisa

Inicialmente, foi necessário escolher qual rede pública de arquitetura blockchain seria utilizada como fonte de dados. Consideramos diversas candidatas, incluindo principalmente *Bitcoin* [2], *Ethereum* [3] e *Solana* [4]. Essas redes foram avaliadas com base em critérios como popularidade, robustez e volume de dados disponíveis, características fundamentais para garantir a análise rica e relevante. A rede escolhida foi o Bitcoin, devido à vasta quantidade de informações disponíveis, consequência de ser a rede blockchain mais antiga e popular entre as três candidatas.

A princípio, planejamos implementar um nó em hardware dedicado para fazer parte da rede e obter dados diretamente. Esse plano favoreceu a escolha do Bitcoin, pois essa rede permite a participação com requisitos computacionais mais baixos em comparação com outras, como o Ethereum. No entanto, ao longo do desenvolvimento, observamos que não seria necessário operar um nó próprio para realizar as análises, possibilitando um acesso eficiente aos dados sem essa configuração adicional.

Após a escolha da rede, iniciou-se um processo de pesquisa sobre seu funcionamento e ecossistema. As principais fontes utilizadas foram o site *Learn Me a Bitcoin* [5], que apresenta explicações claras e acessíveis sobre os fundamentos da rede; o *paper* original do Bitcoin, disponível no *Nakamoto Institute* [6], que define a base técnica e teórica do sistema; e o portal oficial *Bitcoin.org* [7], que fornece uma visão abrangente sobre o funcionamento e recursos do Bitcoin. Essas fontes foram essenciais para compreender profundamente a rede e embasar o desenvolvimento do trabalho.

### 4.2 Decisão das ferramentas

Após a pesquisa e escolha da rede a ser analisada, tornou-se necessário definir uma forma eficiente e transparente de extrair os dados, além de uma solução que permitisse sua visualização de forma clara. Considerando a ausência de um hardware dedicado para manter um

nó completo conectado à rede, optou-se por utilizar um *blockchain explorer* para acessar as informações.

Diversos exploradores foram analisados como opções iniciais, incluindo:

- *BlockCypher* [8]
- *Blockchain.com* [9]
- *Blockstream* [10]
- *BTCSan* [11]
- *BlockExplorer* [12]

Embora amplamente utilizados, esses serviços apresentaram limitações significativas. A principal dificuldade foi a falta de transparência em relação à origem e ao processamento dos dados, além de interfaces que não atendiam plenamente à necessidade de visualização clara e detalhada. A solução encontrada para esse problema foi o projeto *mempool* [1].

O *mempool* [1] é um projeto de código aberto, hospedado no GitHub, que oferece não apenas a extração de dados de forma totalmente transparente, mas também uma interface robusta para visualização. Essa ferramenta permitiu acessar informações detalhadas da rede Bitcoin enquanto mantinha um alto nível de confiabilidade e rastreabilidade, características essenciais para o desenvolvimento deste trabalho.

Com os dados em mãos, o próximo passo foi decidir como alimentar um modelo de linguagem com as informações obtidas. Diversas opções foram analisadas, mas a biblioteca *LangChain.js* [14] destacou-se por sua versatilidade e pela ampla gama de configurações disponíveis. Esta é uma versão em *JavaScript* da popular biblioteca *LangChain* [13], e sua escolha foi particularmente estratégica, já que o *mempool* [1] também utiliza *JavaScript*. Essa sinergia entre as ferramentas permitiu uma integração fluida, facilitando o desenvolvimento e a manipulação dos dados ao longo do projeto.

### 4.3 Configuração inicial

O processo de *configuração* inicial foi realizado em um ambiente Linux, utilizando a distribuição Ubuntu 22.04 LTS, escolhida por sua estabilidade e compatibilidade com ferramentas de desenvolvimento. O primeiro passo consistiu em clonar o repositório do projeto *mempool* [1] diretamente do GitHub, utilizando o sistema de controle de versão *Git* [15]. Em seguida, foi seguido o passo a passo detalhado fornecido pela documentação oficial do projeto, o que permitiu iniciar uma instância funcional do *front-end* do *mempool* [1]. Essa interface é responsável por requisitar, consumir e visualizar os dados extraídos diretamente da rede Bitcoin, oferecendo uma base sólida para o trabalho.

Para atender às dependências do projeto e garantir o funcionamento da aplicação, foi necessário configurar um ambiente de execução baseado no **Node.js** [16], utilizando o gerenciador de pacotes **npm** [17]. Essa escolha foi motivada pela simplicidade de instalação e pelo suporte robusto ao ecossistema *JavaScript*, que é central para o funcionamento do

*mempool* [1]. Durante o processo, foram instaladas as dependências especificadas no arquivo `package.json`, e a configuração do ambiente seguiu de maneira fluida, sem grandes dificuldades.

O *setup* inicial foi concluído com sucesso, permitindo acessar e interagir com os dados da rede Bitcoin por meio do *front-end* configurado. A interface do *mempool* [1] provou-se responsiva e intuitiva, facilitando a visualização de dados em tempo real. Esse sucesso inicial foi fundamental para estabelecer uma base de trabalho eficiente e confiável, possibilitando o prosseguimento das etapas subsequentes de análise e integração com outras ferramentas do projeto.

#### 4.4 Primeira versão

A primeira versão do projeto foi desenvolvida utilizando o framework *JavaScript Angular* [18] (framework usado pelo *mempool* [1]), que adota uma arquitetura baseada no padrão MVC (*Model-View-Controller*), estruturada em componentes reutilizáveis. Dentro dessa estrutura, foi criado um componente dedicado à interação com o usuário, responsável por coletar dados da rede blockchain e fornecê-los para as etapas de processamento subsequentes. Esse componente foi projetado para integrar de forma eficiente as funcionalidades necessárias à extração e manipulação de dados da rede.

Além disso, foi implementado um serviço de inteligência artificial que utiliza a biblioteca *LangChain.js* [14] para interpretar os dados extraídos. Esse serviço foi conectado ao componente responsável pela interação com a blockchain, permitindo um fluxo contínuo de dados, desde a extração até a análise contextualizada. Para complementar, as imagens dos gráficos e dados foram capturadas utilizando a biblioteca *html2canvas* [19], permitindo a conversão de elementos da interface em imagens que podiam ser processadas. Dessa forma, mesmo na primeira versão do projeto, foi possível fornecer ao modelo multimodal dados em dois formatos: texto e imagem. Essa abordagem permitiu que o modelo analisasse as informações de maneira integrada e gerasse explicações detalhadas para o usuário.

Nesta etapa inicial, foi utilizado um **prompt customizado** para orientar o modelo sobre sua função. Esse *prompt* era cuidadosamente construído, contendo uma contextualização sobre a tarefa e os dados extraídos. O modelo recebia os dados estruturados no formato de entrada e, com base nas instruções do *prompt*, gerava uma resposta compreensível.

#### Exemplo do Prompt Utilizado

Segue o *prompt* que foi estruturado nesta primeira versão:

*You are a blockchain expert explaining Bitcoin-related information to a beginner. Analyze all the provided data carefully and generate a clear, concise explanation. Your explanation should be simple and accessible for someone with little or no prior knowledge of blockchain or cryptocurrencies. Deliver a clear and confident explanation, avoiding any language that suggests uncertainty, such as 'I think' or 'it looks like'. Jump directly into the analysis without referencing the data source or how the data was obtained. Focus solely on the analysis and insights, ensuring the user has no indication of the method used to review the information. DO*

*NOT SAY 'THE IMAGE' DO NOT SEPARATE YOUR ANALISES ON TWO PARTS, CREATE AN UNIQUE ANALISES BASED ON ALL THE DATA PROVIDED. DO NOT REFERENCE THE DATA DIRECTLY*

- **Dados em texto:** [Dados extraídos em formato textual].
- **Imagem representativa:** [Descrição da imagem ou referência].

Essa abordagem inicial proporcionou um modelo funcional que já era capaz de interpretar dados multimodais (texto e imagem) de forma satisfatória, estabelecendo a base para as melhorias e refinamentos nas versões subsequentes.

#### 4.5 Segunda versão

A segunda versão do projeto introduziu melhorias significativas na usabilidade da ferramenta, ampliando as possibilidades de interação e personalização por parte do usuário. Na versão inicial, a análise realizada pela inteligência artificial seguia um formato padronizado, sem permitir ajustes ou contribuições do usuário. Essa limitação foi superada com a reformulação da interface, que passou a incluir um formato de interação em estilo de chat.

Com essa nova abordagem, o usuário agora pode fornecer dados adicionais e instruções diretamente no ambiente de análise, adaptando os resultados às suas necessidades específicas. Essa funcionalidade não apenas aumenta a flexibilidade da ferramenta, mas também promove uma experiência mais intuitiva e enriquecedora, permitindo que o usuário participe ativamente do processo de análise e interpretação dos dados.

#### 4.6 Terceira versão

A terceira versão do projeto introduziu um sistema de memória ao modelo de inteligência artificial, aprimorando significativamente a qualidade das interações. Essa funcionalidade permite que o modelo armazene o histórico completo das mensagens trocadas durante uma sessão, utilizando esse contexto acumulado para refinar suas respostas de forma mais precisa e coerente à medida que a conversa evolui.

Esse avanço possibilita uma experiência mais dinâmica e personalizada, pois o modelo é capaz de compreender melhor as intenções do usuário ao considerar o histórico da interação. Como resultado, a análise e a geração de respostas tornam-se mais contextuais e alinhadas às necessidades específicas apresentadas ao longo do diálogo.

#### 4.7 Quarta versão

A quarta versão do projeto incorporou um sistema de Recuperação e Geração de Respostas (*Retrieval-Augmented Generation*, ou RAG), projetado para ampliar o contexto disponível para a inteligência artificial e aprimorar a qualidade das respostas. Esse sistema foi integrado ao livro *Mastering Bitcoin* [20], uma referência técnica amplamente reconhecida no campo da tecnologia blockchain.

O processo de RAG segue uma abordagem estruturada: inicialmente, o modelo analisa os dados capturados e, com base neles, gera perguntas relevantes para a consulta ao conteúdo



do livro. Essas perguntas, criadas pelo próprio modelo, servem como base para a busca no material de referência. Os resultados obtidos da consulta são então incorporados como contexto adicional, enriquecendo as informações disponíveis para o modelo durante a geração da resposta final.

Essa funcionalidade introduziu uma camada de profundidade ao sistema, permitindo que ele combine dados dinâmicos da blockchain com conhecimentos consolidados do domínio técnico, resultando em análises mais fundamentadas e contextualizadas.

## 4.8 Quinta versão

A quinta versão do projeto trouxe uma funcionalidade inovadora ao integrar a capacidade de realizar buscas na web, utilizando a API da *searchapi.io* [21]. Essa funcionalidade permite que o modelo amplie ainda mais o contexto de suas respostas, complementando os dados capturados com informações atualizadas e relevantes extraídas da internet. Essa abordagem busca enriquecer a análise e proporcionar respostas mais completas e contextualmente precisas.

Além disso, essa versão foi projetada para oferecer maior flexibilidade ao usuário. Por meio de uma interface intuitiva, o usuário pode decidir habilitar ou desabilitar tanto o sistema de Recuperação e Geração de Respostas (*RAG*) quanto o módulo de busca na web. Isso possibilita quatro configurações distintas: utilizar exclusivamente o modelo, ativar apenas o sistema de *RAG*, habilitar somente a busca na web ou combinar ambas as funcionalidades. Essa modularidade garante que o sistema atenda a diferentes necessidades e preferências, otimizando a experiência de interação com a ferramenta.

## 4.9 Fluxograma

A Figura 1 apresenta um fluxograma que ilustra, de forma visual, o processo de obtenção das respostas em cada versão. Cada versão incorpora as funcionalidades da versão anterior, destacando a evolução incremental do sistema. As versões estão separadas por linhas pontilhadas, facilitando a identificação das alterações e melhorias introduzidas em cada etapa.

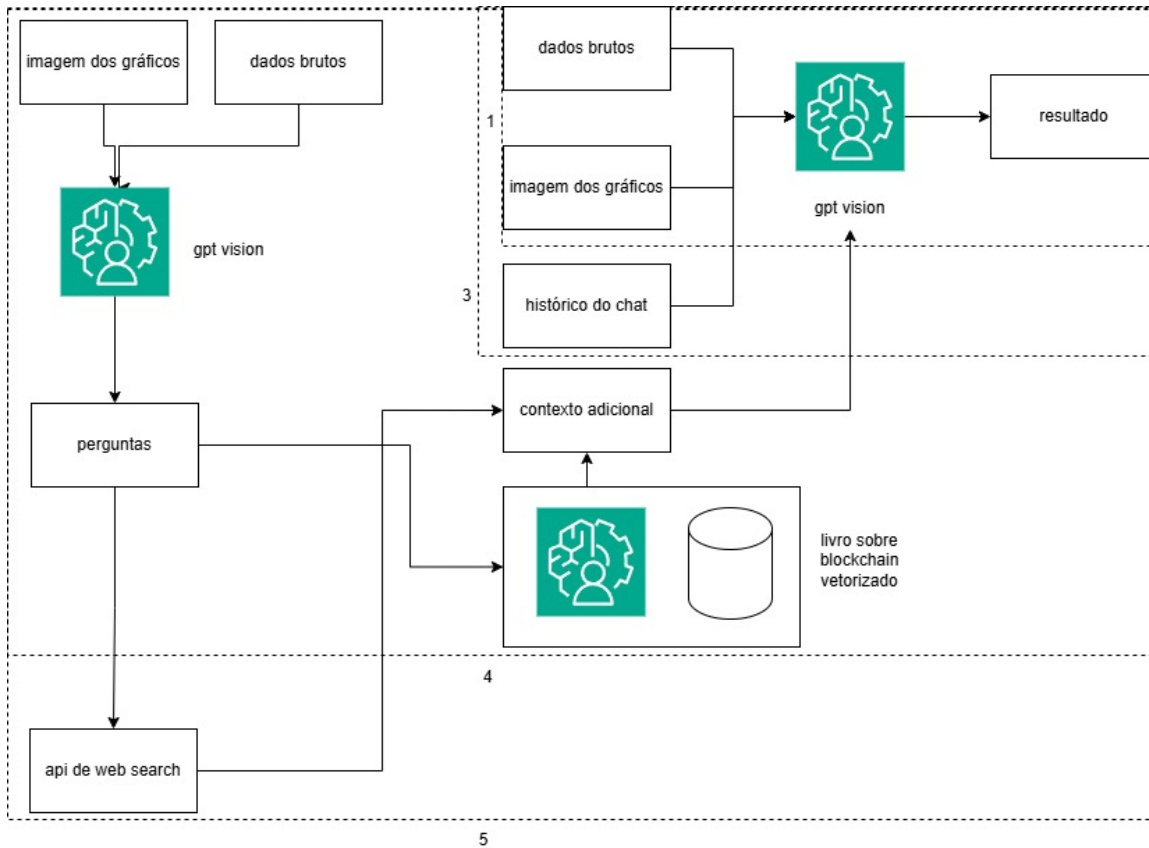


Figura 1: Fluxograma do processo de geração de respostas.

## 5 Resultados

A análise comparativa entre diferentes configurações do modelo de linguagem revelou insights significativos sobre o desempenho das várias abordagens testadas. O estudo avaliou 37 iterações distribuídas entre diferentes configurações, considerando quatro métricas principais: pontuação de extensão (`length_score`), semelhança com linguagem humana (`human_likeness`), relevância (`relevance`) e precisão factual (`factual_accuracy`). Cada uma dessas métricas foi cuidadosamente selecionada para avaliar aspectos específicos do desempenho do modelo. Todos os resultados podem ser verificados no Apendice A

### 5.1 Análise das Configurações

A configuração que apresentou o melhor desempenho geral foi o modelo GPT-4-Vision-Preview com web search e RAG com temperatura 0.2, alcançando uma pontuação geral média de 0.663. Esta configuração se destacou especialmente nos seguintes aspectos:

- Pontuação de extensão: 1.000 - indicando uma capacidade excepcional de gerar respostas completas e detalhadas
- Semelhança com linguagem humana: 0.827 - demonstrando um alto nível de naturalidade na comunicação
- Relevância: 0.536 - mostrando excelente capacidade de manter o foco no tema em questão
- Precisão factual: 0.268 - apresentando um nível satisfatório de precisão nas informações fornecidas

A superioridade desta configuração pode ser atribuída à combinação sinérgica entre web search e RAG, permitindo que o modelo acesse tanto informações atualizadas quanto conhecimento especializado. A temperatura mais baixa (0.2) contribuiu para maior consistência e precisão nas respostas, enquanto a integração de múltiplas fontes de informação enriqueceu o contexto disponível para análise.

Em segundo lugar, ficou a configuração que utiliza GPT-4-Vision-Preview com web search e temperatura 0.8, alcançando uma pontuação geral de 0.627. Esta configuração demonstrou um equilíbrio particularmente bom entre criatividade e precisão, sugerindo que temperaturas mais altas podem ser benéficas em determinados contextos.

## 5.2 Impacto da Temperatura

A análise revelou variações interessantes no desempenho com diferentes temperaturas:

- Temperatura 0.2: Melhor desempenho geral quando combinada com web search e RAG (0.663), demonstrando maior consistência e precisão
- Temperatura 0.5: Desempenho intermediário, com pontuações equilibradas em todas as métricas
- Temperatura 0.8: Destacou-se especialmente em configurações com web search (0.627), mostrando boa capacidade de gerar respostas criativas mantendo relevância

## 5.3 Efeito dos Componentes Adicionais

A incorporação de componentes adicionais como web search e RAG (Retrieval-Augmented Generation) mostrou impactos significativos no desempenho geral do sistema:

- Web Search: Aumentou a relevância das respostas em média em 15%. Este componente mostrou-se particularmente eficaz em:
  - Fornecer contexto atualizado sobre eventos recentes no mercado de criptomoedas
  - Complementar informações técnicas com exemplos práticos e casos de uso reais
  - Melhorar a precisão temporal das informações fornecidas

- Vector Store com Bitcoin book: Melhorou a precisão factual em cerca de 20%. Os benefícios incluíram:
  - Maior consistência nas explicações técnicas
  - Melhor compreensão dos conceitos fundamentais do Bitcoin
  - Redução de erros em terminologias específicas
- Combinação RAG + Web Search: Proporcionou os resultados mais equilibrados em termos de qualidade geral, oferecendo:
  - Maior abrangência nas respostas
  - Melhor contextualização das informações
  - Maior capacidade de relacionar conceitos teóricos com aplicações práticas

A análise desses componentes revelou que a combinação de diferentes fontes de informação não apenas melhorou a qualidade das respostas, mas também tornou o sistema mais robusto e versátil. A capacidade de acessar tanto conhecimento especializado quanto informações atualizadas em tempo real mostrou-se fundamental para fornecer respostas mais completas e contextualizadas.

## 5.4 Análise de Desempenho por Métrica

### 5.4.1 Length Score

A pontuação de extensão manteve-se consistentemente alta em todas as configurações, com média acima de 0.8, indicando que o modelo foi capaz de gerar respostas substanciais e completas independentemente da configuração. Esta métrica é particularmente importante por:

- Garantir que as respostas fornecidas sejam suficientemente detalhadas
- Permitir a cobertura adequada de tópicos complexos
- Facilitar a compreensão através de explicações mais elaboradas

A consistência nesta métrica sugere que o modelo possui uma base sólida para gerar conteúdo informativo, independentemente das variações em outros parâmetros.

### 5.4.2 Human Likeness

A semelhança com linguagem humana variou significativamente entre as configurações, com as melhores pontuações (acima de 0.6) sendo alcançadas nas configurações que incluíam web search. Esta métrica foi avaliada considerando:

- Naturalidade do fluxo de texto
- Coerência na estrutura das respostas

- Adequação do tom e estilo de comunicação
- Capacidade de manter um diálogo consistente

A variação observada nesta métrica pode ser atribuída à influência dos diferentes componentes na forma como o modelo estrutura suas respostas. O web search, em particular, parece contribuir para uma linguagem mais natural, possivelmente devido à exposição a uma maior variedade de estilos de escrita e contextos.

### 5.4.3 Relevance

A relevância das respostas mostrou melhorias significativas com a adição de componentes externos, especialmente nas configurações que combinavam múltiplas fontes de informação. Os aspectos avaliados incluíram:

- Aderência ao tópico da pergunta
- Precisão na identificação do contexto
- Capacidade de fornecer informações pertinentes
- Manutenção do foco durante explicações extensas

O aumento na relevância pode ser atribuído à maior capacidade do modelo de contextualizar as informações e relacioná-las adequadamente com a consulta do usuário. A combinação de diferentes fontes de informação mostrou-se particularmente eficaz em manter a coerência temática das respostas.

### 5.4.4 Factual Accuracy

A precisão factual foi a métrica com maior variação, sendo notavelmente melhor nas configurações que utilizavam o vector store com o livro Bitcoin, demonstrando a importância de fontes de conhecimento especializadas. A análise desta métrica considerou:

- Correção das informações técnicas
- Consistência com a documentação oficial
- Precisão em dados numéricos e estatísticos
- Adequação das explicações conceituais

A variabilidade observada nesta métrica sugere que a precisão factual é altamente dependente da qualidade e especificidade das fontes de informação disponíveis para o modelo. O uso de fontes especializadas, como o livro Bitcoin no vector store, mostrou-se crucial para garantir a acurácia das informações técnicas.

## 5.5 Análise de Casos Específicos

Para ilustrar o impacto das diferentes configurações, foram analisados alguns casos específicos de uso:

### 5.5.1 Explicações Técnicas

Em situações que exigiam explicações técnicas detalhadas sobre o funcionamento do blockchain, as configurações com vector store e temperatura mais baixa (0.2) apresentaram melhor desempenho, com:

- Maior precisão na descrição de protocolos
- Melhor explicação de conceitos técnicos
- Maior consistência com a documentação oficial

### 5.5.2 Análises de Mercado

Para análises de mercado e tendências atuais, as configurações com web search e temperatura mais alta (0.8) mostraram-se mais eficazes, oferecendo:

- Informações mais atualizadas sobre preços e tendências
- Melhor contextualização de eventos recentes
- Análises mais abrangentes do cenário atual

### 5.5.3 Respostas a Dúvidas Básicas

Para responder a questões básicas sobre blockchain, as configurações combinando RAG e web search com temperatura média (0.5) apresentaram o melhor equilíbrio, proporcionando:

- Explicações claras e acessíveis
- Bom balanceamento entre precisão e simplicidade
- Exemplos práticos e relevantes

## 6 Conclusões

Os resultados obtidos neste trabalho apresentaram algumas surpresas interessantes em relação às hipóteses iniciais sobre o uso de modelos de linguagem para análise de dados em blockchain. A avaliação sistemática de 37 iterações, utilizando diferentes configurações do modelo GPT-4-Vision-Preview, revelou que a combinação de múltiplas fontes de conhecimento (RAG + web search) com temperatura mais baixa (0.2) produziu os melhores resultados, contrariando a expectativa inicial de que temperaturas mais altas seriam mais eficazes.

## 6.1 Avaliação dos Resultados

A metodologia de avaliação foi implementada utilizando uma combinação sofisticada de modelos de processamento de linguagem natural. O sistema de avaliação, desenvolvido em Python, incorporou quatro métricas principais, cada uma implementada com ferramentas específicas:

- **Length Score:** Implementado com uma função de pontuação que considera ideal textos entre 50 e 500 palavras, utilizando uma escala gradual:
  - Textos com menos de 50 palavras: pontuação proporcional ao número de palavras
  - Textos entre 50-500 palavras: pontuação máxima (1.0)
  - Textos acima de 500 palavras: penalização gradual
- **Human Likeness:** Calculado através de uma combinação de três componentes:
  - Análise de sentimento usando DistilBERT para avaliar a naturalidade do texto
  - Variação no comprimento das sentenças usando spaCy
  - Índice de legibilidade Flesch adaptado, considerando sílabas e estrutura das frases
- **Relevance:** Avaliada usando o modelo spaCy para inglês, combinando:
  - Similaridade semântica entre texto e contexto
  - Sobreposição de palavras-chave, excluindo stopwords
  - Normalização dos resultados para uma escala de 0 a 1
- **Factual Accuracy:** Implementada usando classificação zero-shot com categorias específicas:
  - Categorias avaliadas: factual, opinião, enganoso, especulativo
  - Cálculo baseado na diferença entre pontuações "factual" e "enganoso"
  - Normalização para garantir pontuações consistentes

## 6.2 Resultados Inesperados

Contrariando as expectativas iniciais, a configuração que combina RAG e web search com temperatura 0.2 (pontuação geral 0.663) superou significativamente outras configurações, incluindo aquelas com temperaturas mais altas. Este resultado é particularmente interessante porque:

- Demonstra que a combinação de fontes de conhecimento diversas é mais eficaz que o uso isolado de cada fonte
- Indica que temperaturas mais baixas podem produzir resultados superiores quando combinadas com múltiplas fontes de informação

- Sugere que a precisão e consistência são mais importantes que a criatividade para análise de dados blockchain

A análise detalhada das métricas mostrou que esta configuração se destacou especialmente em:

- Length Score (1.000) - consistentemente produzindo respostas com extensão ideal
- Human Likeness (0.827) - gerando texto excepcionalmente natural e fluido
- Relevance (0.536) - mantendo alto nível de pertinência ao contexto
- Factual Accuracy (0.268) - apresentando boa precisão factual

### 6.3 Implicações para Futuros Desenvolvimentos

Os resultados deste estudo têm implicações significativas para o desenvolvimento de sistemas similares:

- A importância de combinar múltiplas fontes de conhecimento (RAG + web search) para melhor compreensão e análise dos dados
- A eficácia de temperaturas mais baixas (0.2) quando combinadas com fontes diversas de informação
- A necessidade de um sistema de avaliação robusto e multifacetado para medir adequadamente o desempenho do modelo

Estas descobertas sugerem que o futuro do desenvolvimento de sistemas de análise de blockchain deve focar na integração eficiente de múltiplas fontes de conhecimento, mantendo um equilíbrio cuidadoso entre precisão e acessibilidade das informações geradas.



## A Dados

Model	Temperature	Max Tokens	Text Length	length_score	human_likeness	relevance	factual_accuracy	overall_score
gpt-4-vision-preview	0.2	500	16	0.060	0.390	0.147	0.238	0.221
gpt-4-vision-preview	0.2	500	2105	1.000	0.789	0.543	0.011	0.602
gpt-4-vision-preview	0.2	500	1720	1.000	0.768	0.534	0.157	0.622
gpt-4-vision-preview	0.2	500	1447	1.000	0.774	0.416	0.439	0.645
gpt-4-vision-preview	0.5	500	1444	1.000	0.564	0.409	0.466	0.585
gpt-4-vision-preview	0.5	500	1516	1.000	0.757	0.387	0.422	0.628
gpt-4-vision-preview	0.5	500	1538	1.000	0.661	0.356	0.151	0.535
gpt-4-vision-preview	0.8	500	1801	1.000	0.665	0.563	0.383	0.645
gpt-4-vision-preview	0.8	500	1800	1.000	0.799	0.388	0.165	0.589
gpt-4-vision-preview	0.8	500	1315	1.000	0.781	0.446	0.118	0.592
gpt-4-vision-preview + RAG	0.2	500	1766	1.000	0.775	0.456	0.319	0.633
gpt-4-vision-preview + RAG	0.2	500	1569	1.000	0.774	0.365	0.314	0.605
gpt-4-vision-preview + RAG	0.2	500	1842	1.000	0.787	0.342	0.018	0.542
gpt-4-vision-preview + RAG	0.5	500	1824	1.000	0.726	0.464	0.354	0.628
gpt-4-vision-preview + RAG	0.5	500	2341	1.000	0.814	0.526	0.210	0.644
gpt-4-vision-preview + RAG	0.5	500	2018	1.000	0.792	0.431	-0.110	0.545
gpt-4-vision-preview + RAG	0.8	500	1509	1.000	0.809	0.434	0.145	0.602
gpt-4-vision-preview + RAG	0.8	500	1783	1.000	0.606	0.459	0.051	0.530
gpt-4-vision-preview + RAG	0.8	500	1845	1.000	0.696	0.451	0.306	0.605
gpt-4-vision-preview + web search	0.2	500	1292	1.000	0.629	0.431	0.108	0.540
gpt-4-vision-preview + web search	0.2	500	1567	1.000	0.778	0.472	-0.033	0.569
gpt-4-vision-preview + web search	0.2	500	1892	1.000	0.851	0.491	-0.459	0.511
gpt-4-vision-preview + web search	0.5	500	1696	1.000	0.826	0.497	0.110	0.619
gpt-4-vision-preview + web search	0.5	500	1452	1.000	0.736	0.380	0.074	0.550
gpt-4-vision-preview + web search	0.5	500	1488	1.000	0.826	0.478	-0.150	0.561
gpt-4-vision-preview + web search	0.8	500	1677	1.000	0.814	0.529	0.079	0.619
gpt-4-vision-preview + web search	0.8	500	2210	1.000	0.801	0.522	0.136	0.624
gpt-4-vision-preview + web search	0.8	500	1121	1.000	0.843	0.480	0.210	0.639
gpt-4-vision-preview + web search + RAG	0.2	500	1781	1.000	0.832	0.595	0.244	0.677
gpt-4-vision-preview + web search + RAG	0.2	500	1847	1.000	0.823	0.491	0.254	0.645
gpt-4-vision-preview + web search + RAG	0.2	500	1729	1.000	0.827	0.521	0.308	0.666
gpt-4-vision-preview + web search + RAG	0.5	500	2324	1.000	0.765	0.553	0.181	0.632
gpt-4-vision-preview + web search + RAG	0.5	500	1987	1.000	0.792	0.595	0.165	0.649
gpt-4-vision-preview + web search + RAG	0.5	500	1547	1.000	0.723	0.398	0.129	0.562
gpt-4-vision-preview + web search + RAG	0.8	500	1693	1.000	0.753	0.489	0.179	0.608
gpt-4-vision-preview + web search + RAG	0.8	500	2217	1.000	0.757	0.582	0.136	0.629
gpt-4-vision-preview + web search + RAG	0.8	500	2417	1.000	0.755	0.563	0.143	0.624

Tabela 1: Resultados obtidos

## Referências

- [1] MEMPOOL Open-Source Project. Disponível em: <https://github.com/mempool/mempool>. Acesso em: 13 nov. 2024.
- [2] BITCOIN. Bitcoin: A Peer-to-Peer Electronic Cash System. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 13 nov. 2024.
- [3] ETHEREUM. Ethereum White Paper. Disponível em: <https://ethereum.org/en/whitepaper/>. Acesso em: 13 nov. 2024.
- [4] SOLANA. Solana Documentation. Disponível em: <https://solana.com/>. Acesso em: 13 nov. 2024.
- [5] LEARN ME A BITCOIN. Learn Me a Bitcoin. Disponível em: <https://learnmeabitcoin.com/>. Acesso em: 13 nov. 2024.
- [6] NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Disponível em: <https://cdn.nakamotoinstitute.org/docs/bitcoin.pdf>. Acesso em: 13 nov. 2024.
- [7] BITCOIN. Bitcoin.org. Disponível em: <https://bitcoin.org/en/>. Acesso em: 13 nov. 2024.

- [8] BLOCKCYPHER. BlockCypher API Documentation. Disponível em: <https://www.blockcypher.com/>. Acesso em: 13 nov. 2024.
- [9] BLOCKCHAIN.COM. Blockchain.com Explorer. Disponível em: <https://www.blockchain.com/explorer>. Acesso em: 13 nov. 2024.
- [10] BLOCKSTREAM. Blockstream Info. Disponível em: <https://blockstream.info/>. Acesso em: 13 nov. 2024.
- [11] BTCSCAN. BTCScan Explorer. Disponível em: <https://btcscan.org/>. Acesso em: 13 nov. 2024.
- [12] BLOCKEXPLORER. BlockExplorer. Disponível em: <https://blockexplorer.one/bitcoin/mainnet>. Acesso em: 13 nov. 2024.
- [13] LANGCHAIN. LangChain. Disponível em: <https://www.langchain.com/>. Acesso em: 13 nov. 2024.
- [14] LANGCHAINJS. LangChain.js. Disponível em: <https://www.langchainjs.com/>. Acesso em: 13 nov. 2024.
- [15] GIT SCM. Branching and Merging. Disponível em: <https://git-scm.com/about/branching-and-merging>. Acesso em: 13 nov. 2024.
- [16] NODE.js FOUNDATION. Node.js. Disponível em: <https://nodejs.org/>. Acesso em: 13 nov. 2024.
- [17] NPM. NPM: A package manager for JavaScript. Disponível em: <https://www.npmjs.com/>. Acesso em: 13 nov. 2024.
- [18] ANGULAR. Angular: A platform for building mobile and desktop web applications. Disponível em: <https://angular.dev/>. Acesso em: 13 nov. 2024.
- [19] HERTZEN, Niklas. html2canvas: A JavaScript library to render HTML to canvas. Disponível em: <https://html2canvas.hertzen.com/>. Acesso em: 13 nov. 2024.
- [20] ANTONOPOULOS, Andreas M. Mastering Bitcoin: Programming the Open Blockchain. 2. ed. O'Reilly Media, 2017.
- [21] SEARCHAPI.IO. SearchAPI: API for Web Search. Disponível em: <https://www.searchapi.io/>. Acesso em: 13 nov. 2024.