



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
INSTITUTO METRÓPOLE DIGITAL  
PROGRAMA DE RESIDÊNCIA EM TECNOLOGIA DA INFORMAÇÃO

**Exploração de técnicas de engenharia de prompt para aprimorar os  
resultados do uso de LLM no TCMRio**

Jefferson Rodrigues do Nascimento

Natal-RN, Brasil

2024

Jefferson Rodrigues do Nascimento

**Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de LLM no TCMRio**

Trabalho de Conclusão de Curso apresentado ao Programa de Residência em Tecnologia da Informação do Instituto Metrópole Digital da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Tecnologia da Informação. Área de Concentração: Desenvolvimento de Software.

Orientador: Prof. Dra. Isabel Dillmann Nunes

Natal-RN, Brasil

2024

Universidade Federal do Rio Grande do Norte - UFRN  
Sistema de Bibliotecas - SISBI  
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Nascimento, Jefferson Rodrigues do.

Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de LLM no TCMRio / Jefferson Rodrigues do Nascimento. - 2024.

64 f.: il.

Monografia (especialização) - Universidade Federal do Rio Grande do Norte, Instituto Metr pole Digital, Resid ncia em Tecnologia da Informa o, Natal, RN, 2024.

Orienta o: Profa. Dra. Isabel Dillmann Nunes.

1. Engenharia de Prompt. 2. Intelig ncia artificial. 3. Processamento de linguagem natural. 4. LangChain. I. Nunes, Isabel Dillmann. II. T tulo.

RN/UF/BCZM

CDU 004.8

Jefferson Rodrigues do Nascimento

## Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de LLM no TCMRio

Trabalho de Conclusão de Curso  
apresentado ao Programa de Residência em  
Tecnologia da Informação Aplicada à Área  
Jurídica do Instituto Metr pole Digital da  
Universidade Federal do Rio Grande do  
Norte como requisito parcial para a  
obtenção do t tulo de Especialista em  
Tecnologia da Informa o.  rea de  
Concentra o: Desenvolvimento de  
Software .

Trabalho aprovado. Natal-RN, Brasil, vinte e oito de fevereiro de dois mil e vinte e quatro:

---

Prof. Dra. Isabel Dillmann Nunes

Orientador

---

Prof. Dr. Jair Cavalcanti Leite

Examinador

---

Prof. Dr. Eduardo Henrique da Silva Aranha

Examinador

Natal-RN, Brasil  
2024

## **Agradecimentos**

Em primeiro lugar, gostaria de expressar minha mais profunda gratidão a Deus, por me guiar, fortalecer e iluminar o caminho durante esta jornada. Sem Sua presença constante e Sua graça, nada disso teria sido possível.

À minha querida esposa, Juliana Rodrigues, minha eterna companheira, cujo amor, paciência e apoio incondicional foram o farol que me guiou através dos momentos de incerteza e desafio. Sua fé inabalável em mim e sua capacidade de ver a luz mesmo nos dias mais escuros foram fundamentais para minha perseverança e sucesso.

À minha família, que me ofereceu um porto seguro e o suporte emocional necessário para enfrentar os obstáculos que surgiram no caminho. Agradeço pelo incentivo constante, pelas palavras de motivação e, principalmente, pelo amor incondicional que me fortaleceu em cada etapa desta jornada.

Não posso deixar de expressar minha sincera gratidão à minha orientadora, Isabel Nunes, cuja sabedoria, dedicação e orientação meticulosa foram essenciais para o desenvolvimento e a conclusão deste trabalho. Sua paciência, expertise e incentivo foram cruciais para superar as dificuldades técnicas e intelectuais enfrentadas durante a realização deste trabalho.

A todos que direta ou indiretamente contribuíram para a realização deste trabalho, meu sincero agradecimento. Cada palavra de encorajamento, cada gesto de apoio, cada momento compartilhado, todos foram vitais para tornar esta jornada não apenas possível, mas também significativa.

Encerro esta seção de agradecimentos com um sentimento de gratidão imensa e a certeza de que as lições aprendidas e as experiências compartilhadas durante a elaboração deste trabalho permanecerão comigo por toda a vida. Obrigado a todos que fizeram parte desta etapa tão importante do meu caminho acadêmico e pessoal.

*"O temor do SENHOR é o princípio do saber, mas os loucos desprezam a sabedoria e o ensino." - Provérbios 1:7*

## Resumo

A adoção de Modelos de Linguagem de Grande Escala (LLMs) pelo Tribunal de Contas do Município do Rio de Janeiro (TCMRio) marca um avanço significativo na sua jornada de inovação tecnológica, visando a melhoria das operações internas através da Inteligência Artificial. A motivação deste estudo origina-se de uma prova de conceito realizada pelo TCMRio, a qual revelou limitações significativas na qualidade das interações automatizadas, evidenciando a necessidade de explorar novas abordagens para otimizar esses resultados. Neste cenário, a pesquisa visa explorar técnicas de engenharia de prompt como uma solução para superar os desafios identificados, potencializando a eficácia dos LLMs empregados. Os objetivos específicos incluem a revisão teórica acerca dos fundamentos da Inteligência Artificial (IA), Processamento de Linguagem Natural (PLN), e engenharia de prompt, com ênfase no aprimoramento do uso de LLMs, particularmente o GPT, no ambiente do TCMRio. Adicionalmente, propõe-se a exploração prática dessas técnicas através do desenvolvimento e teste de prompts que visam melhorar a qualidade e a relevância das respostas dos chatbots. A metodologia adotada caracteriza-se por uma abordagem mista, combinando análise bibliográfica detalhada e aplicação prática. Esta envolve a implementação de um protótipo de chatbot utilizando LLM, seguido de uma série de testes iterativos para refinar os prompts com base nas técnicas de engenharia de prompt estudadas. A avaliação dos resultados focou na eficiência e eficácia das respostas fornecidas pelo chatbot, comparando-as com as da prova de conceito inicial. Os resultados alcançados demonstram a viabilidade e efetividade das técnicas de engenharia de prompt na melhoria significativa da qualidade das interações com o chatbot. As estratégias desenvolvidas contribuem para um caminho promissor para a implementação futura de soluções baseadas em IA que atendam às necessidades específicas do TCMRio.

*Palavras-chave:* Engenharia de Prompt. Inteligência Artificial. Processamento de Linguagem Natural. LangChain.

## **Abstract**

The adoption of Large Language Models (LLMs) by the Court of Auditors of the Municipality of Rio de Janeiro (TCMRio) signifies a significant leap in its technological innovation journey, aiming to enhance internal operations through Artificial Intelligence. The motivation for this study stems from a proof of concept conducted by TCMRio, which uncovered substantial limitations in the quality of automated interactions, highlighting the need to explore new approaches to optimize these outcomes. In this context, the research seeks to investigate prompt engineering techniques as a solution to overcome the identified challenges, thereby enhancing the effectiveness of the employed LLMs. The specific objectives include a theoretical review of the fundamentals of Artificial Intelligence (AI), Natural Language Processing (NLP), and prompt engineering, with an emphasis on improving the use of LLMs, particularly GPT, within the TCMRio environment. Additionally, it proposes the practical exploration of these techniques through the development and testing of prompts aimed at improving the quality and relevance of chatbot responses. The adopted methodology features a mixed approach, combining detailed bibliographic analysis and practical application. This involves the implementation of a chatbot prototype using LLM, followed by a series of iterative tests to refine the prompts based on the studied prompt engineering techniques. The evaluation of results focused on the efficiency and effectiveness of the chatbot's responses, comparing them to those from the initial proof of concept. The findings demonstrate the viability and effectiveness of prompt engineering techniques in significantly improving the quality of interactions with the chatbot. The developed strategies contribute to a promising path for the future implementation of AI-based solutions that meet the specific needs of TCMRio.

*Keywords:* Prompt Engineering. Artificial Intelligence. Natural Language Processing. LangChain.

## **Lista de ilustrações**

Figura 1 – Etapas para geração de texto pelos modelos de linguagem	26
Figura 2 – Figura código RedisVectorStore	43
Figura 3 – Figura de fluxo de embedding dos dados para serem armazenados	44
Figura 4 – Estrutura do prompt	45
Figura 5 – Fluxo de chat implementado com LangChain	46
Figura 6 – Configuração dos chunks	47

## **Lista de quadros**

Quadro 1 – Linhas de pensamento para conceituar inteligência artificial.

18

## **Lista de tabelas**

Tabela 1 - Perguntas originais da Poc TCMRio e com engenharia de prompt.	48
Tabela 2 - Respostas originais da Poc TCMRio e respostas do Chat LangChain.	52

## **Lista de abreviaturas e siglas**

IA	Inteligência Artificial
IMD	Instituto Metr�pole Digital
GPT	Generative Pretrained Transformer
PLN	Processamento de Linguagem Natural
TCMRio	Tribunal de Contas do Munic�pio do Rio de Janeiro
TCE-SP	Tribunal de Contas do Estado de S�o Paulo
LLM	Large Language Model(modelos de linguagem de grande escala)
GQA	Generative Question-Answering
MLGE	Modelos de Linguagem de Grande Escala
BERT	Bidirectional Encoder Representations from Transformers
UFRN	Universidade Federal do Rio Grande do Norte

## Lista de símbolos

$\Gamma$	Letra grega Gama
$\Lambda$	Lambda
$\zeta$	Letra grega minúscula zeta

## Sumário

<b>1 Introdução</b>	<b>15</b>
1.1 Objetivos	16
1.2 Organização do Trabalho	16
<b>2 Referencial Teórico</b>	<b>18</b>
2.1 Inteligência Artificial	18
2.2 Processamento de Linguagem Natural (PLN)	21
2.3 LLM	24
2.4 ChatGPT	27
2.5 Engenharia de Prompt	28
2.6 LangChain: Biblioteca para trabalhar com LLMs	36
<b>3 Metodologia</b>	<b>42</b>
3.1 Definição do Problema	42
3.2 Desenvolvimento do Chat	42
3.2.1 Framework LangChain	43
3.2.2 Banco Redis	43
3.2.3 Api OpenAI	44
3.2.4 Prompt	44
<b>4. Resultados</b>	<b>48</b>
4.1 Implementação de Testes	48
4.2 Desafios Identificados	60
4.3 Conclusões e Implicações para Futuras Pesquisas	60
<b>Referências</b>	<b>62</b>

# 1 Introdução

A Inteligência Artificial (IA) e o Processamento de Linguagem Natural (PLN) têm revolucionado a maneira como interagimos com máquinas, proporcionando avanços significativos na criação de sistemas que entendem, interpretam e respondem à linguagem humana de maneira inteligente. Segundo Vaswani et al. (2017), os modelos de linguagem generativa, como o GPT (Generative Pre-trained Transformer), representam um marco nessa evolução, oferecendo uma abordagem inovadora para a geração de texto automatizada que simula a fluência e a profundidade da comunicação humana. No entanto, a eficácia desses sistemas em aplicações práticas depende substancialmente da qualidade dos prompts que recebem, o que destaca a importância da engenharia de prompt como um campo de estudo emergente.

A engenharia de prompt envolve a elaboração estratégica de instruções ou estímulos que guiam os modelos de IA a gerar respostas específicas, precisas e relevantes. Como indicado por Brown et al. (2020), essa prática se tornou fundamental para aproveitar ao máximo o potencial dos modelos de linguagem generativa, especialmente em aplicações de chatbots e sistemas interativos baseados em texto. Nesse contexto, a LangChain, uma biblioteca que facilita a integração de capacidades de PLN com cadeias de ferramentas de IA, emerge como uma solução promissora para desenvolver e otimizar essas interações.

Este trabalho se insere nesse panorama inovador com o objetivo de explorar o desenvolvimento e a otimização de um chatbot utilizando LangChain e técnicas avançadas de engenharia de prompt centradas no modelo GPT. A relevância desse estudo é amplificada pela crescente demanda por interfaces de usuário mais naturais, inteligentes e adaptativas, capazes de entender e processar linguagem humana de forma eficaz.

A metodologia adotada inclui uma análise aprofundada da literatura existente sobre IA, PLN, e modelos de linguagem generativa, além da implementação prática de um protótipo de chatbot. Este protótipo é submetido a uma série de testes iterativos para refinar os prompts e avaliar o desempenho do sistema em termos de precisão das respostas e satisfação do usuário. Através desta abordagem, busca-se não apenas contribuir para o avanço teórico e metodológico no campo da engenharia de prompt, mas também fornecer insights valiosos para a aplicação prática de chatbots baseados em IA.

A análise dos resultados permitirá identificar não só as potencialidades mas também os desafios inerentes à engenharia de prompt, propondo soluções e melhorias para aprimorar a comunicação entre humanos e máquinas. Diante do exposto, o escopo deste trabalho abrange desde a concepção teórica até a implementação prática de um sistema de chat inteligente, oferecendo uma contribuição significativa para o campo da Inteligência Artificial e do Processamento de Linguagem Natural. Ao final, espera-se que os resultados deste estudo forneçam uma base sólida para futuras pesquisas e aplicações práticas envolvendo chatbots e tecnologias de IA generativa.

## **1.1 Objetivos**

O objetivo geral deste trabalho é desenvolver um guia que detalhe o uso de técnicas de engenharia de prompt para aumentar a eficiência das ferramentas de Inteligência Artificial Generativa no Tribunal de Contas do Município do Rio de Janeiro (TCMRio), visando aprimorar a análise documental.

Como forma de atingir o objetivo geral deste trabalho, tem-se os seguintes objetivos específicos:

- a) Investigar como o uso de modelos de linguagem de grande escala (LLM) pode ser aplicado para melhorar a eficácia na análise de documentos e na geração de respostas relevantes no contexto do Tribunal de Contas.
- b) Desenvolver uma Solução de Chat com o framework LangChain que permita o teste e a experimentação com modelos de prompts.
- c) Realizar testes aplicando técnicas de Engenharia de Prompt utilizando as perguntas da prova de conceito do TCMRio.

## **1.2 Organização do Trabalho**

Este trabalho está organizado em seções que conduzem o leitor através da jornada de pesquisa e desenvolvimento empreendida. Após esta introdução, serão apresentados os fundamentos teóricos que sustentam a investigação, seguidos pela metodologia adotada para o desenvolvimento e teste de modelos de prompts. Posteriormente, será realizada uma análise dos resultados alcançados, comparando-os com os estados anteriores à implementação das

técnicas de engenharia de prompt. Finalmente, o trabalho conclui com uma reflexão sobre as implicações dos achados para a prática de análise documental no TCMRio e sugestões para futuras pesquisas na área.

## 2 Referencial Teórico

Para contextualizar a discussão, iniciaremos explorando os princípios da Inteligência Artificial, os conceitos relacionados à engenharia de prompt, examinando suas práticas e desafios de implementação e refinamento de modelos de prompt. Também será discutida uma abordagem externa ao TCMRio de case de sucesso, o ANIA (TCE-SP).

### 2.1 Inteligência Artificial

Segundo Russell e Norvig (2013), "A IA é baseada em algoritmos e modelos matemáticos que permitem que os sistemas aprendam com dados e experiências passadas. Esses sistemas são capazes de analisar grandes quantidades de dados em tempo real, identificar padrões e tomar decisões com base em informações precisas e confiáveis."

O quadro 1 está categorizado em duas dimensões principais: processos de pensamento e comportamento, e comparação com desempenho humano versus ideal de racionalidade.

As definições focadas em processos de pensamento e raciocínio se alinham com o desempenho humano, enquanto as orientadas para comportamento se relacionam com a racionalidade ideal, ou seja, a capacidade de agir corretamente com base no conhecimento disponível.

Quadro 1 – Linhas de pensamento para conceituar inteligência artificial.

<b>Pensando como um humano</b>	<b>Pensando racionalmente</b>
<p>“O novo e interessante esforço para fazer os computadores pensarem (...) <i>máquinas com mentes</i>, no sentido total e literal.” (Haugeland, 1985)</p> <p>“[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...” (Bellman, 1978)</p>	<p>“O estudo das faculdades mentais pelo uso de modelos computacionais.” (Charniak e McDermott, 1985)</p> <p>“O estudo das computações que tornam possível perceber, raciocinar e agir.” (Winston, 1992)</p>
<b>Agindo como seres humanos</b>	<b>Agindo racionalmente</b>
<p>“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990)</p> <p>“O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas.” (Rich and Knight, 1991)</p>	<p>“Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (Poole <i>et al.</i>, 1998)</p> <p>“AI... está relacionada a um desempenho inteligente de artefatos.” (Nilsson, 1998)</p>

Fonte: Russel e Norvig (2013)

Essa diversidade de abordagens revela a complexidade e a riqueza do campo da inteligência artificial. Russell e Norvig também mostram no quadro 1, que a Inteligência Artificial tem sido estudada sob quatro perspectivas: Agindo de forma humana, pensando de forma humana, pensando racionalmente e agindo racionalmente.

**Agindo de forma humana, o teste de Turing:** Proposto por Alan Turing, o Teste de Turing avalia se um computador pode ser considerado inteligente ao ser indistinguível de um humano em suas respostas a perguntas escritas. Para passar neste teste, um computador precisa de habilidades como processamento de linguagem natural, raciocínio automatizado e aprendizado de máquina. O "teste de Turing total" expande isso para incluir visão computacional e robótica, mas a IA tem se concentrado mais em entender a inteligência do que em passar no teste.

**Pensando de forma humana, modelagem Cognitiva:** Esta abordagem busca entender e replicar o processo de pensamento humano. Utilizando métodos como introspecção, experimentos psicológicos e imagens cerebrais, busca-se desenvolver programas de computador que imitem a maneira como os humanos pensam e resolvem problemas. A ciência cognitiva, que une a IA e a psicologia experimental, trabalha para formar teorias precisas sobre os processos mentais humanos.

**Pensando racionalmente, “leis do Pensamento”:** Baseando-se na lógica aristotélica, essa abordagem visa desenvolver sistemas de inteligência artificial que utilizem o raciocínio lógico para resolver problemas. Enfrenta desafios como a complexidade de expressar conhecimento informal em termos lógicos e o alto custo computacional para resolver problemas práticos.

**Agindo racionalmente, agente Racional:** Enfoca na ação para alcançar o melhor resultado possível. Um agente racional na IA é projetado para operar de forma autônoma, perceber seu ambiente e tomar decisões baseadas na lógica e no conhecimento adquirido. Esta abordagem é ampla e cientificamente viável, embora a racionalidade perfeita seja um objetivo desafiador em ambientes complexos (RUSSELL e NORVIG, 2013).

O termo Inteligência Artificial (IA) abrange uma gama diversificada de aplicações que conseguem realizar tarefas complexas, anteriormente reservadas à interação humana, como interagir com clientes online ou jogar xadrez. Essa designação é comumente utilizada de forma intercambiável com seus subcampos, incluindo machine learning.

No entanto, é crucial compreender as diferenças sutis existentes entre esses conceitos. Por exemplo, o machine learning concentra-se na construção de sistemas que aprendem e aprimoram seu desempenho com base nos dados assimilados. É relevante destacar que, embora todo machine learning seja considerado IA, nem toda IA incorpora elementos de machine learning. Essa distinção reflete a diversidade de abordagens dentro da IA, abrangendo diferentes métodos para alcançar a automação e a execução inteligente de tarefas (ORACLE, 2023).

Para Cozman, Plonski e Neri (2021) Inteligência Artificial é “a área que se ocupa de construir artefatos artificiais que apresentam comportamento inteligente”. Definir comportamento inteligente é uma tarefa complexa, pois a noção de "inteligência" é mutável. Os humanos são surpreendentemente versáteis na forma como interpretam essa ideia, aceitando facilmente a inteligência parcial de determinados animais e ajustando-se sem esforço a tecnologias digitais, mesmo quando estas possuem restrições cognitivas notáveis.

A complexidade de reproduzir um comportamento similar ao humano, reconhecido como inteligente, mas possivelmente não totalmente racional, contrasta com a busca por um comportamento racional baseado em princípios. A popularização do termo "Inteligência Artificial" permeia a literatura técnica e o imaginário popular, impactando diversas áreas como Medicina, Direito e Engenharia.

A sociedade vivencia simultaneamente a expectativa de ganhos em bem-estar e produtividade, bem como a apreensão diante de possíveis cenários apocalípticos relacionados à inteligência artificial. A confusão entre IA e atividades envolvendo aparelhos digitais é destacada, evidenciando que algumas inovações recentes, atribuídas à inteligência artificial, resultam da automatização de tarefas cotidianas ou da aplicação de tecnologias já consolidadas, como exemplificado pelo caso das smart câmeras (COZMAN, PLONSKI e NERI 2021).

## 2.2 Processamento de Linguagem Natural (PLN)

O processamento de linguagem natural (PLN), como um braço da inteligência artificial, utiliza técnicas de machine learning para compreender a estrutura e significado presentes em textos. Este campo abrange tanto o reconhecimento quanto a geração de linguagem natural, destacando-se como uma ferramenta valiosa na interpretação e manipulação de dados textuais.

Um subconjunto crucial do PLN é a compreensão do idioma natural (NLU), que se concentra em entender o verdadeiro significado subjacente aos textos. A NLU desempenha funções importantes, como categorizar, arquivar e analisar o texto de maneira contextualizada. No entanto, o PLN vai além, aspirando não apenas à compreensão, mas também à capacidade de realizar tomadas de decisão com base nesse entendimento mais profundo. Isso proporciona às organizações meios eficazes para analisar textos, extrair informações sobre pessoas, lugares e eventos, e compreender as opiniões presentes em mídias sociais e interações com clientes. Este campo emergente desempenha um papel fundamental na evolução das capacidades das máquinas em entender e interagir com a linguagem humana de maneira mais sofisticada e perspicaz.

No contexto dos chatbots, os métodos de Processamento de Linguagem Natural (PLN) ocupam uma posição fundamental, sendo essenciais para o seu funcionamento contemporâneo. Embora esses sistemas não sejam perfeitos, são capazes de lidar facilmente com tarefas padrão, como informar clientes sobre produtos ou serviços e responder a perguntas. Além do uso em chatbots, o PLN encontra aplicação em várias áreas, como classificação de texto, reconhecimento de caracteres, correção automática, resumo automático, entre outras.

As possibilidades e desafios do PLN incluem a complexidade da transformação de texto em linguagem natural, abordando questões como ambiguidade, sinonímia, correferência e variações no estilo de escrita. O PLN continua a avançar, enfrentando desafios para aprimorar a compreensão semântica, lidar com ambiguidades e capturar a riqueza da linguagem natural. Embora os métodos atuais forneçam soluções eficazes, a busca por abordagens mais aprimoradas e avançadas permanece constante, impulsionando a evolução contínua dessa área crucial da inteligência artificial.

O Processamento de Linguagem Natural tornou-se uma parte intrínseca da vida diária, ganhando ainda mais destaque à medida que a tecnologia da linguagem é incorporada a diversos setores. No varejo, por exemplo, o PLN é aplicado em chatbots de atendimento ao cliente, proporcionando interações mais eficientes. Na área médica, a interpretação e resumo de registros eletrônicos de saúde são aprimorados pela aplicação da tecnologia de processamento de linguagem natural.

Além disso, gigantes da tecnologia incorporam o PLN para aprimorar os resultados de seus mecanismos de busca, enquanto redes sociais a utilizam para detectar e filtrar discursos de ódio. Embora o PLN esteja alcançando níveis crescentes de sofisticação, ainda enfrenta desafios significativos. Os sistemas atuais tendem a apresentar preconceitos, incoerências e, por vezes, comportamento inconsistente. Apesar dessas limitações, os engenheiros de aprendizado automático têm vastas oportunidades para aplicar o PLN de maneiras cada vez mais essenciais, contribuindo para uma sociedade mais funcional e conectada (IBM, 2023).

Para Kovacs (2023) “As melhores tecnologias de detecção de spam usam os recursos de classificação de texto do processamento de linguagem natural para verificar e-mails em busca de palavras que geralmente indicam spam ou phishing”.

Existem três níveis de reconhecimento na técnica de processamento de linguagem natural, esses níveis compreendem a análise morfológica, sintática e semântica.

Na análise morfológica, as palavras são minuciosamente examinadas para identificar sua forma básica (lematização) e características gramaticais, como parte do discurso, gênero, número, tempo verbal, entre outros. A análise sintática utiliza a estrutura gramatical para compreender as relações entre as palavras e a organização das frases. Por sua vez, a análise semântica busca compreender o significado das palavras e frases, considerando o contexto em que são empregadas.

Destaca-se a ambiguidade como uma característica comum na linguagem natural, levando à necessidade de uma etapa de desambiguação. O objetivo aqui é resolver ambiguidades e determinar o significado mais apropriado com base no contexto, possibilitando a extração de informações específicas, como nomes de pessoas, datas, localizações e eventos importantes. Após as análises, muitos sistemas de PLN recorrem ao Machine Learning. Essa abordagem visa ajustar modelos matemáticos aos dados para realizar

tarefas preditivas. Exemplos práticos incluem a classificação de texto, análise de sentimentos e tradução, demonstrando a aplicação eficaz do Machine Learning no contexto do PLN (GONÇALVES, 2023).

As Técnicas de Processamento de Linguagem Natural, envolvem a divisão de texto ou fala humana em partes compreensíveis para programas de computador. Técnicas comuns de processamento e análise de texto no PLN incluem:

**Marcação de Parte do Discurso:** Nesse processo, o software de PLN identifica e marca palavras individuais em uma frase conforme seu uso contextual, categorizando-as como substantivos, verbos, adjetivos ou advérbios. Isso auxilia o computador a compreender as relações significativas entre as palavras.

**Desambiguação do Sentido da Palavra:** Diante da ambiguidade de algumas palavras em contextos distintos, o PLN realiza a desambiguação para determinar o significado específico da palavra, seja por meio do treinamento do modelo de idioma ou referenciando definições em dicionários.

**Reconhecimento de Voz:** A transformação de dados de voz em texto, realizada pelo reconhecimento de fala, envolve a análise de sotaques, entonação, insultos e uso não padrão da gramática em conversas cotidianas.

**Tradução Automática:** Utilizando processamento de linguagem natural, o software de tradução automática converte texto ou fala de um idioma para outro, mantendo a precisão contextual.

**Reconhecimento de Entidade Nomeada:** Identificando nomes exclusivos para pessoas, lugares, eventos e empresas, o PLN utiliza o reconhecimento de entidade nomeada para determinar as relações entre diferentes entidades em uma frase. A resolução de referências pode aprimorar essa análise, estabelecendo conexões entre palavras distintas que se referem à mesma entidade.

**Análise de Sentimento:** Baseada em inteligência artificial, a análise de sentimentos no PLN busca interpretar as emoções transmitidas por dados textuais. O software analisa o texto em busca de palavras ou frases que revelem sentimentos como insatisfação, felicidade, dúvida ou arrependimento.

Os benefícios do Processamento de Linguagem Natural são amplos, proporcionando vantagens tanto para empresas quanto para consumidores. Seja na tradução instantânea de textos entre idiomas ou na geração de insights de negócios por meio da análise de sentimentos em inúmeras avaliações, o PLN desempenha um papel significativo.

## 2.3 LLM

Um *Large Language Model* (LLM) representa um modelo de inteligência artificial que passa por extenso treinamento em uma grande quantidade de textos. Essencialmente, esses modelos desenvolvem a habilidade de antecipar a próxima palavra em uma frase, fundamentando-se nas palavras anteriores, resultando na geração de texto coeso.

Esta capacidade dos LLMs tem implicações significativas, podendo ser aplicada em diversas áreas, desde a criação de textos que mimetizam a escrita humana até a produção autônoma de conteúdo coerente e contextualmente relevante. A integração de LLMs redefine a interação entre humanos e tecnologia, abrindo portas para aplicações inovadoras em processamento de linguagem natural e produção textual automatizada.

Segundo Magalhães (2023) "As tecnologias de inteligência artificial de conversação, como o ChatGPT e o Bard, dependem de uma estrutura chamada LLM. Ele é capaz de acumular todo o conhecimento das bases de dados e convertê-lo em respostas". Por entender e gerar texto de forma convincente, o Large Language Model pode ser aplicado a vários cenários diferentes.

O LLM pode ser adaptado para funções específicas. A inclusão de novos prompts durante as interações possibilita a incorporação de informações adicionais à base de dados do LLM, resultando em respostas mais precisas. Quanto à aquisição de contexto, o LLM se alimenta de dados públicos, mas é possível aprimorar ainda mais suas capacidades por meio da contextualização, onde cada novo prompt fornece dados adicionais para a IA, contribuindo para resultados mais refinados e adaptados às necessidades do usuário.

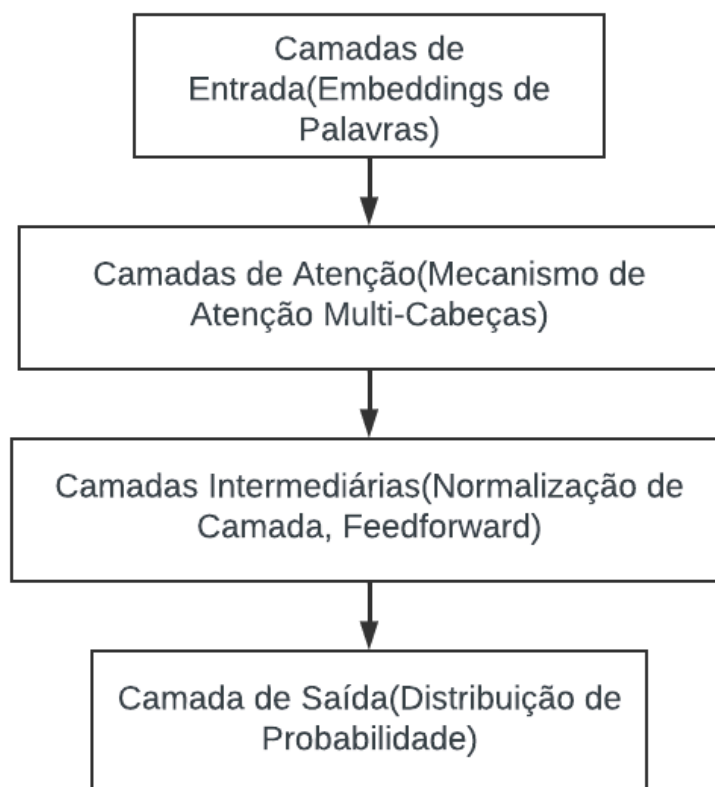
Nas interações com o usuário o contexto que o LLM vai adquirindo não fica salvo. Depois de uma sequência de prompts a Inteligência artificial encerra a conversa e os próximos diálogos acabam iniciando a partir da base que o LLM já tem, um exemplo disso é a IA do

Bing que tem suas informações restauradas a cada seis prompts, limitando a retenção de contexto durante a interação.

É importante ressaltar que mesmo apresentando um grande potencial, surgem preocupações em relação ao treinamento desses modelos de IA e à criação de conteúdos originais. Algumas plataformas podem replicar conteúdo existente, resultando em plágio, e há casos em que a IA interpreta um texto já existente como se o tivesse inventado. Além disso, a questão dos direitos autorais surge quando as linguagens coletam dados de obras com proteção intelectual, levando a processos legais. Também existe o risco de as plataformas de IA gerarem e divulgarem informações falsas como fatos, adicionando mais uma preocupação ao cenário.

Os modelos de linguagem transformam simples entradas de texto em saídas complexas e significativas, este processo é facilitado por uma série de camadas e mecanismos especializados, cada um desempenhando um papel fundamental na interpretação e geração de linguagem. A seguir, uma imagem ilustrativa oferece uma visão detalhada dessas etapas, desde a conversão inicial do texto em representações matemáticas até a seleção final das palavras que formarão a resposta do modelo. Essa representação visual serve como uma porta de entrada para o entendimento dos componentes críticos que permitem aos modelos de linguagem entender e produzir linguagem humana com uma precisão cada vez maior.

Figura 1 – Etapas para geração de texto pelos modelos de linguagem.



Fonte: Elaborado pelo autor

**Camadas de Entrada:** Representação visual das entradas de texto sendo transformadas em embeddings de palavras, que são vetores que capturam o significado das palavras.

**Camadas de Atenção:** Ilustração de como as camadas de atenção ponderam diferentes partes da entrada para focar em informações relevantes para cada palavra durante o processamento.

**Mecanismo de Atenção Multi-Cabeças:** Destaque para o mecanismo de atenção multi-cabeças, mostrando que o modelo analisa a entrada de várias perspectivas simultaneamente.

**Camadas Intermediárias:** Desenho esquemático das várias camadas intermediárias de processamento, incluindo normalização de camada e mecanismos de feedforward.

Camada de Saída: Representação da saída do modelo, geralmente na forma de uma distribuição de probabilidade sobre o vocabulário, indicando a próxima palavra mais provável.

## 2.4 ChatGPT

O ChatGPT é um chatbot baseado em inteligência artificial que interage com humanos de maneira realista por meio de texto. Ele gera respostas fluidas e naturais, utilizando um extenso banco de textos da internet, incluindo artigos, notícias e posts. Treinado com a técnica de aprendizado de máquina chamada "Aprendizagem por Reforço com Feedback Humano" (RLHF), o ChatGPT melhora suas respostas com base nos feedbacks dos usuários, tornando-as mais rápidas, precisas e inteligentes.

O chatbot também enfrentou controvérsias em relação aos direitos autorais, levantando questões sobre a propriedade de materiais gerados por IA e sua participação em prêmios e concursos. A polêmica do plágio surgiu quando a IA imitou o estilo de artistas humanos, resultando em processos de escritores que alegavam treinamento ilegal com obras não autorizadas.

A inteligência artificial do ChatGPT possui também o potencial de redefinir a prática da advocacia, tornando-a mais eficiente. No entanto, é imperativo adotar precauções adequadas para evitar possíveis complicações. O papel do ChatGPT na configuração de um novo panorama profissional é tão significativo quanto os riscos associados ao seu uso sem as devidas precauções. Aspectos cruciais como segurança e ética no emprego do chat na advocacia demandam atenção especial. Apesar de sua utilidade nesta área, é essencial compreender algumas estratégias para empregar a ferramenta de forma eficaz. Além disso, é crucial analisar se essa tecnologia representa uma substituição inevitável para os advogados ou se é uma ferramenta construtiva.

Apesar das promessas de eficiência e produtividade, o uso do ChatGPT sem precauções adequadas pode acarretar riscos significativos. Existem casos reais de advogados nos Estados Unidos que enfrentaram problemas ao dependerem cegamente das respostas geradas pela IA. Tais perigos incluem informações desatualizadas, falta de contexto ou interpretação equivocada, conteúdo genérico e respostas tendenciosas. Isso ressalta a importância de verificar e validar as informações fornecidas pela inteligência artificial.

Apesar dos perigos, o ChatGPT pode ser uma ferramenta valiosa quando utilizado de maneira ética. Existem três formas éticas de incorporá-lo na advocacia: assistência no estudo de teses e jurisprudência, revisão e análise de documentos, e auxílio na redação jurídica. É crucial enfatizar a importância de checar e confirmar as informações fornecidas pela IA, garantindo que ela contribua de maneira eficaz para a prática jurídica (STRAZZI, 2023).

## 2.5 Engenharia de Prompt

Com a popularização de novas plataformas de desenvolvimento de conteúdo, especialmente no campo de inteligência artificial generativa, como ChatGPT<sup>1</sup>, Google Bard<sup>2</sup>, Bing Chat<sup>3</sup> e Midjourney<sup>4</sup>. Esses serviços auxiliam na criação de textos, ilustrações, tabelas, gráficos e aplicativos, buscando soluções criativas para tarefas simples. No entanto, destaca-se a importância de não apenas enviar pedidos simples para essas IAs, mas de elaborar comandos precisos, completos e bem escritos para obter resultados mais aprimorados.

A prática nessas plataformas deu origem à engenharia de prompt, uma especialidade que se tornou uma profissão presente, envolvendo a criação de prompts eficazes para direcionar chatbots a fornecerem respostas mais precisas e úteis. A engenharia de prompt é definida como a arte e ciência de criar instruções na caixa de entrada de uma IA generativa, visando orientar a produção de conteúdo de maneira específica e eficiente.

Engenharia de prompt é um conjunto de técnicas e métodos para projetar, escrever e otimizar instruções para Modelos de Linguagem Generativos, chamados prompts, de modo que as respostas do modelo sejam precisas, concretas, replicáveis e factualmente corretas. Os prompts são consideradas uma forma de programação, pois podem personalizar as saídas e interações com um LLM, envolvendo a adaptação de instruções em linguagem natural para obter respostas desejadas, garantindo resultados contextualmente precisos e aumentando a utilidade de modelos de linguagem generativos em várias aplicações, como educação médica, radiologia e educação científica.

---

<sup>1</sup> ChatGPT. Disponível em: <https://openai.com/chatgpt> . Acesso em: 24 jan. 2024

<sup>2</sup> Google Bard. Disponível em: <https://bard.google.com> . Acesso em: 24 jan. 2024

<sup>3</sup> Bing Chat. Disponível em: <https://www.bing.com/chat> . Acesso em: 24 jan. 2024

<sup>4</sup> Midjourney. Disponível em: <https://www.midjourney.com> . Acesso em: 24 jan. 2024

Segundo Ferreira (2023) “A Engenharia por Prompt é um processo extremamente necessário para que a IA-Generativa produza respostas adequadas ao usuário.”

A engenharia de prompt inclui a padronização de terminologia e conceitos, com várias classificações de prompts surgindo com base em critérios diferentes. De acordo com a estrutura, prompts podem ser formulados usando perguntas abertas ou fechadas. Perguntas abertas permitem respostas mais extensas e detalhadas, enquanto perguntas fechadas têm respostas específicas e limitadas.

Para KLEINA (2023) “A complexidade da explicação em forma de ordem, o uso de termos-chave e até a capacidade de fazer correções ao longo da produção do chatbot são algumas habilidades adquiridas por essa especialização da engenharia de prompt.”

Prompts podem ser categorizados em níveis de 1 a 4, variando de perguntas simples a prompts mais complexos que incluem exemplos, permitindo que o modelo de linguagem descompunha a solicitação em componentes individuais. Além disso, prompts podem ser classificados como instrutivos, de sistema, pergunta-resposta ou mistos, dependendo da abordagem utilizada.

Reprodutibilidade é uma característica desejada, embora os LLMs produzam respostas aleatórias devido ao seu design intrínseco. O desenvolvimento de prompts é reconhecido como um processo iterativo, e é essencial que o texto do prompt seja claro, conciso e evite complexidade desnecessária. Por isso é importante usar técnicas de engenharia de prompt como:

**Prompting Explícito e Direto (Explicit and Direct Prompting):** Formular perguntas ou instruções de forma clara e específica para obter respostas diretas e focadas.

**Prompting Baseado em Exemplos (Example-based Prompting):** Fornecer exemplos para guiar o modelo de IA na produção de respostas em um formato ou estilo específico.

**Prompting de Corrente de Pensamento (Chain-of-Thought Prompting):** Encorajar a IA a explicar o processo de pensamento passo a passo para resolver problemas complexos. O Prompting com corrente de pensamento é uma técnica que divide uma questão complexa em partes lógicas, imitando uma linha de pensamento. Isso ajuda o modelo a resolver problemas em etapas intermediárias, promovendo o raciocínio.

**Prompting Zero-shot:** Fazer perguntas ou comandos sem fornecer exemplos prévios, confiando na capacidade inerente do modelo de compreender e responder.

**Prompting de Auto-Indução (Self-Induction Prompting):** Pedir ao modelo que gere suas próprias instruções ou regras como parte da resposta.

**Reformulação de Prompts (Prompt Reframing):** Modificar a formulação de um prompt para melhorar a precisão ou relevância da resposta.

**Prompts Negativos/Positivos (Negative/Positive Prompting):** Incluir instruções específicas sobre o que deve ou não ser incluído nas respostas.

**Prompting Híbrido (Hybrid Prompting):** Combinar diferentes técnicas de prompt para maximizar a eficácia.

**Prompting de Contextualização (Contextualization Prompting):** Adicionar informações relevantes ao prompt para ajudar o modelo a entender melhor o contexto.

**Prompting Criativo (Creative Prompting):** Utilizar abordagens criativas, como analogias ou cenários hipotéticos, para estimular respostas inovadoras.

**Prompting de Escalada (Escalation Prompting):** Aumentar gradualmente a complexidade do prompt para explorar temas ou ideias mais profundas.

**Prompting de Desambiguação (Disambiguation Prompting):** Especificar claramente no prompt para reduzir ambiguidades e melhorar a precisão das respostas.

**Prompting Iterativo (Iterative Prompting):** Refinar um prompt com base nas respostas recebidas, ajustando para obter resultados melhores ou mais precisos.

**Prompting para Exploração de Conceitos (Concept Exploration Prompting):** Solicitar à IA que explique ou expanda um conceito específico, auxiliando na compreensão ou geração de novas ideias.

**Prompting para Geração de Ideias (Idea Generation Prompting):** Incentivar a IA a gerar ideias criativas ou soluções para problemas apresentados.

**Prompting de Contraste (Contrast Prompting):** Utilizar dois conceitos, ideias ou cenários opostos para explorar diferenças ou criar uma comparação.

**Prompting com Limitações Específicas (Specific Limitation Prompting):** Definir restrições no prompt, como limites de tempo ou escopo, para guiar as respostas da IA.

**Prompting para Análise Crítica (Critical Analysis Prompting):** Pedir à IA para avaliar ou criticar um conceito, argumento ou obra.

**Prompting Socrático (Socratic Prompting):** Empregar um estilo de questionamento que promova o pensamento crítico e a auto-reflexão.

**Prompting de Role-Playing (Role-Playing Prompting):** Criar cenários onde a IA assume um papel ou persona específica, como um especialista em determinado campo ou um personagem fictício, para explorar diferentes perspectivas ou abordagens em suas respostas.

Prompts mal projetados podem resultar em respostas vagas, tendenciosas, enganosas ou ambíguas, sendo a alucinação um problema significativo. A verificação de fatos nas respostas de LLMs conversacionais é destacada como necessária, especialmente em relação a citações acadêmicas.

A clareza é essencial. Evitar ambiguidades é crucial para prompts eficazes. Instruções diretas asseguram que a IA compreenda a tarefa, evitando confusões para o usuário. Exemplo: Em vez de "Conte-me sobre a história", prefira "Descreva os eventos mais significativos durante a Revolução Francesa".

Criar uma persona para a IA é poderoso. Isso melhora a compreensão do contexto e tom desejados. A persona define o estilo das respostas, alinhando-as com as expectativas do usuário. Exemplo: Se a IA adota uma persona amigável, um prompt apropriado seria: "Compartilhe sua opinião sobre o filme como se estivesse conversando com um amigo."

O formato do prompt é vital. Especificar o formato desejado da resposta guia a IA na geração de conteúdo adequado. Esclarecer se a resposta deve ser uma lista, por exemplo,

orienta a IA. Exemplo: Para obter recomendações de leitura, use um prompt como: "Forneça uma lista de livros recomendados sobre inteligência artificial em bullet points."

A imparcialidade é crucial. Evitar influenciar a resposta desejada garante que a IA forneça respostas objetivas e imparciais, sem viés. Exemplo: Evitar conduzir a resposta significa não dar pistas para respostas tendenciosas. Um prompt não tendencioso seria: "Explique os prós e contras da energia solar."

Manter o escopo do prompt conciso é essencial. Instruções curtas e específicas ajudam a evitar confusões, permitindo que a IA se concentre na tarefa em mãos. Exemplo: Em vez de "Fale sobre a história da música clássica", prefira "Descreva a contribuição de Beethoven para a música clássica." (Iara, 2023).

Experimentar várias formulações do prompt ao utilizar a geração de texto para obter resultados mais refinados. Modelos de linguagem, como o GPT-4, aprendem padrões durante o treinamento com grandes conjuntos de texto. A formulação do prompt influencia quais padrões e conhecimentos são ativados no modelo, resultando em gerações de texto distintas. Mesmo pequenas variações na formulação podem levar a interpretações e construções de contexto diferentes. Por exemplo, perguntas semelhantes, como "Quais são os benefícios do exercício físico?" e "Por que devo me exercitar?", podem induzir o modelo a fornecer respostas diferentes devido à sutil mudança na formulação do prompt.

Colocar as instruções importantes no início do prompt ao trabalhar com modelos de linguagem como o GPT. Utilizar símbolos como ### ou "" para separar a instrução do contexto ajuda a destacar claramente essas partes. Isso é eficaz devido ao "decaimento de atenção" nos modelos, onde as informações iniciais têm mais peso. Os símbolos funcionam como marcadores, auxiliando o modelo na diferenciação e categorização, especialmente em prompts complexos com múltiplas partes.

Ao trabalhar com modelos de linguagem, como o GPT-4, é crucial ser específico, descritivo e detalhado no prompt. A precisão é vital, pois o modelo não faz suposições além do que é explicitamente fornecido. Seja claro sobre o contexto desejado, resultado esperado, comprimento, formato e estilo. Especificidades, como detalhes sobre o tópico, comprimento desejado de um resumo, público-alvo e estilo, devem ser incluídas no prompt para obter

respostas mais precisas. A clareza nas instruções é fundamental, pois determina a utilidade do resultado final, mesmo em um modelo sofisticado como o GPT-4.

Ao expressar o formato de saída desejado para modelos de linguagem, como o GPT, é altamente benéfico fornecer exemplos concretos. Modelos respondem efetivamente a exemplos, oferecendo uma compreensão mais precisa das solicitações. Em vez de simplesmente pedir uma tarefa, como extrair entidades de um texto, fornecer exemplos claros do formato desejado, como uma lista separada por vírgulas, aumenta a clareza das instruções. Esses exemplos servem como modelos para o modelo de linguagem seguir, reduzindo mal-entendidos e facilitando a extração programática confiável de múltiplas saídas.

Ao trabalhar com modelos de linguagem, como o GPT, é recomendado começar com a abordagem "zero-shot", na qual o modelo gera respostas sem exemplos explícitos. Essa prática é eficaz para muitas tarefas, sem a necessidade de preparação de exemplos de treinamento. No entanto, se os resultados não forem satisfatórios, pode-se optar pela abordagem "few-shot", onde o modelo recebe alguns exemplos da resposta desejada antes do prompt, melhorando sua compreensão da solicitação. A estratégia "few-shot" envolve o fornecimento de menos de 10 exemplos, adaptando-se conforme necessário para otimizar o desempenho do modelo.

Para otimizar a compreensão do modelo, é aconselhável ser preciso em suas requisições. Evite instruções vagas ou indiretas que possam levar a interpretações ambíguas. Por exemplo, ao solicitar uma descrição de produto curta, fornecer parâmetros claros, como "Use um parágrafo de 3 a 5 frases", aumenta a probabilidade de obter resultados desejados, evitando inconsistências ou imprecisões.

Ao dar instruções a um modelo de linguagem, é mais eficaz concentrar-se no que deseja que ele faça, em vez de apenas definir o que não deve fazer. Embora estabelecer limites seja importante, é igualmente crucial guiar o modelo em ações apropriadas. Por exemplo, em uma simulação de conversa entre um agente e um cliente, em vez de proibir a solicitação de informações pessoais identificáveis, como usuário e senha, é mais eficaz instruir o modelo a diagnosticar o problema, sugerir soluções e orientar o usuário para um recurso de ajuda específico. Essa abordagem proporciona uma orientação clara para a ação desejada, resultando em respostas de maior qualidade e utilidade.

Ao solicitar a geração de código, é benéfico utilizar "palavras-chave" que direcionam o modelo para um padrão específico. Essas palavras-chave funcionam como dicas indicando o tipo de código desejado, orientando eficazmente o modelo. Por exemplo, ao pedir uma função Python, começar com a palavra "import" ajuda o modelo a reconhecer o contexto Python. Da mesma forma, ao usar a palavra "SELECT", o modelo entende que se espera uma instrução SQL. Assim, em vez de instruções vagas, como "Escreva uma função Python simples", é mais eficaz começar com uma palavra-chave, como "import", para guiar o modelo na direção desejada.

Essa abordagem específica otimiza a compreensão do modelo, resultando em códigos mais precisos e adequados ao contexto. Utilizar palavras-chave relevantes no início da solicitação melhora a capacidade do modelo de gerar o código desejado, proporcionando resultados mais coerentes e alinhados com as expectativas do usuário (GADELHA, 2023).

De acordo com White et al. (2023) os padrões de prompt são classificados em cinco categorias: Semântica de Entrada, Personalização de Saída, Identificação de Erro, Melhoria de Prompt e Interação. Cada categoria é resumida a seguir:

- Semântica de Entrada: Trata da compreensão do LLM sobre a entrada e sua tradução em algo útil para gerar saída. Inclui a criação de uma linguagem meta, útil quando a linguagem de entrada padrão não é adequada.
- Personalização de Saída: Foca em restringir ou adaptar tipos, formatos, estruturas ou outras propriedades da saída gerada pelo LLM. Inclui padrões como Automatizador de Saída, Persona, Gerador de Visualização, Receita e Modelo.
- Identificação de Erro: Concentra-se em identificar e resolver erros na saída gerada pelo LLM. Inclui a Lista de Verificação de Fatos e Reflexão.
- Melhoria de Prompt: Foca em melhorar a qualidade da entrada e saída, incluindo o Refinamento de Perguntas, Abordagens Alternativas, Verificador Cognitivo e Quebrador de Recusa.
- Interação: Relaciona-se com a interação entre o usuário e o LLM, incluindo Interação Invertida, Jogo e Geração Infinita.

## Padrões Específicos

- Criação de Linguagem Meta: Permite criar uma linguagem customizada para o LLM entender, útil para expressar ideias específicas.
- Automatizador de Saída: Permite criar scripts para automatizar tarefas sugeridas pelo LLM.
- Persona: Dá ao LLM uma persona ou papel a desempenhar na geração de saída.
- Gerador de Visualização: Permite gerar visualizações produzindo saídas textuais que podem alimentar outras ferramentas.
- Receita: Obtém uma sequência de passos ou ações para alcançar um resultado final desejado.
- Modelo: Permite especificar um modelo para a saída, que o LLM preenche com conteúdo.
- Lista de Verificação de Fatos: Requer que o LLM gere uma lista de fatos que devem ser verificados.
- Reflexão: Requer que o LLM introspekte sua saída e identifique erros.
- Refinamento de Pergunta: Garante que o LLM sugira uma versão melhorada da pergunta do usuário.
- Abordagens Alternativas: Sugere maneiras alternativas de realizar uma tarefa especificada pelo usuário.
- Verificador Cognitivo: Instrui o LLM a sugerir uma série de subperguntas para responder antes de combinar as respostas e produzir uma resposta à pergunta geral.
- Quebrador de Recusa: Requer que o LLM reformule automaticamente a pergunta do usuário quando se recusa a produzir uma resposta.
- Interação Invertida: Requer que o LLM faça perguntas em vez de gerar saída.

- Jogo: Requer que o LLM gere saída na forma de um jogo.
- Geração Infinita: Requer que o LLM gere saída indefinidamente sem que o usuário tenha que reintroduzir o prompt do gerador a cada vez.
- Gerente de Contexto: Permite ao usuário especificar o contexto para a saída do LLM.

A importância do conhecimento profundo no ChatGPT reside na capacidade de refinar consultas para obter respostas mais precisas e relevantes. Compreender a estrutura e limitações do modelo, como seu conhecimento até setembro de 2021 e possíveis vieses, é crucial ao elaborar consultas avançadas. A utilização de conhecimento específico ao formular perguntas contribui para respostas mais úteis e focadas. Incorporar termos técnicos e jargões da área ao buscar informações, como no contexto médico, ajuda na obtenção de dados mais precisos. Diferentes estratégias na formulação de perguntas, como especificidade, uso de termos técnicos e divisão de tópicos complexos, são essenciais para obter respostas mais precisas do ChatGPT.

## **2.6 LangChain: Biblioteca para trabalhar com LLMs**

O LangChain é um framework de código aberto projetado para facilitar o desenvolvimento de aplicativos alimentados por modelos de linguagem, especialmente modelos de linguagem grandes (Large Language Model - LLM). Diferenciando-se das APIs tradicionais, o LangChain é consciente de dados e agentes, permitindo a integração com diversas fontes de dados para experiências mais personalizadas. Além disso, capacita modelos de linguagem para interações dinâmicas com o ambiente. Este framework agiliza o desenvolvimento de aplicações como chatbots, Generative Question-Answering (GQA) e sumarização, possibilitando a criação de aplicativos exclusivos através da interconexão de módulos em torno de um LLM.

O LangChain oferece diversos módulos que possibilitam a construção de aplicativos de modelo de linguagem. Esses módulos podem ser combinados para criar desde aplicativos simples até aplicações mais complexas. O bloco de construção mais fundamental envolve a

chamada de um modelo de linguagem para obter previsões com base em uma entrada específica.

Além disso, o LangChain permite a criação de cadeias (chains), que são combinações de primitivos como LLMs ou outras cadeias, organizadas em fluxos de trabalho de várias etapas. Essas cadeias podem ser formadas por diferentes tipos de módulos, sendo um exemplo essencial o LLMChain, que consiste em um PromptTemplate e um LLM. Essa abordagem permite a construção de aplicações mais sofisticadas, onde a entrada do usuário é formatada por um PromptTemplate e, em seguida, passada para um modelo de linguagem para geração de respostas.

Em resumo, o LangChain proporciona flexibilidade na construção de aplicativos de modelo de linguagem, permitindo a combinação de módulos para atender a diferentes necessidades e a criação de fluxos de trabalho complexos através das cadeias (REIS, 2023).

Conforme discutido por Topsakal e Akinci (2020), o desenvolvimento de aplicações baseadas em Modelos de Linguagem de Grande Escala pode ser significativamente acelerado com o uso do LangChain. Projetado para facilitar o desenvolvimento de aplicações que utilizam Modelos de Linguagem de Grande Escala (MLGEs), tais como modelos avançados, exemplificados pelo BERT (Bidirectional Encoder Representations from Transformers) do Google e GPT (Generative Pretrained Transformer) da OpenAI, e T5 (Text-to-Text Transfer Transformer) do Google. Essa estrutura oferece uma série de componentes modulares e pipelines customizáveis, permitindo que desenvolvedores integrem facilmente diversas fontes de dados e interajam com uma ampla variedade de aplicações.

O objetivo do LangChain é simplificar o uso de MLGEs em desenvolvimentos específicos, promovendo uma abordagem mais acessível e eficiente na criação de aplicações de inteligência artificial. Entre seus principais componentes estão os Prompts, Memória, Cadeias e Agentes, que juntos possibilitam a geração de respostas textuais inteligentes, contextualizadas e de alta qualidade.

No coração do LangChain estão os "Prompts", que servem como entrada para os MLGEs. Estes são dinamicamente gerados e podem incluir desde a entrada do usuário até exemplos específicos e instruções detalhadas para o modelo, visando otimizar a geração de respostas. Através do uso de "Templates de Prompts", o LangChain permite a criação de

prompts reproduzíveis que podem ser customizados de acordo com os parâmetros fornecidos pelo usuário final. Isso é especialmente útil em aplicações que necessitam de interações complexas e específicas com o usuário, oferecendo um nível de flexibilidade e precisão na comunicação com os MLGEs.

Os componentes de "Memória" e "Cadeias" do LangChain tratam, respectivamente, do armazenamento de conversas anteriores e da orquestração de operações sequenciais em textos ou dados. Enquanto a memória ajuda a criar uma ilusão de continuidade nas interações com o modelo, as cadeias permitem combinar diferentes blocos de construção, como MLGEs e prompts, para executar tarefas complexas. Isso inclui desde responder a perguntas com base em documentos até a geração de conteúdo multimídia sintético, exemplificando a versatilidade do LangChain em lidar com uma vasta gama de desafios de processamento de linguagem natural.

Outro aspecto importante do LangChain é a implementação de "Agentes", que são utilizados para determinar a sequência de ferramentas e operações baseadas nas entradas dos usuários. Isso introduz uma camada adicional de flexibilidade e inteligência no desenvolvimento de aplicações, permitindo que o sistema decida autonomamente qual ferramenta ou sequência de ações é mais adequada para cada situação. Essa abordagem baseada em agentes é fundamental para criar aplicações que necessitam de adaptação e tomada de decisões em tempo real, abrindo novas possibilidades para a personalização e eficiência das soluções de IA.

O LangChain se destaca por seu papel em democratizar o acesso e a utilização de tecnologias de IA avançadas. Ao fornecer uma estrutura que simplifica a integração de MLGEs em aplicações customizadas, ele não apenas acelera o desenvolvimento de soluções inteligentes, mas também incentiva a inovação em diversos campos, como educação, saúde e atendimento ao cliente. Com o LangChain, desenvolvedores podem explorar o potencial completo dos Modelos de Linguagem de Grande Escala, criando aplicações mais inteligentes, contextuais e interativas que antes eram difíceis ou demoradas para serem desenvolvidas.

O LangChain simplifica o desenvolvimento de aplicativos ao permitir a referência a dados próprios em um armazenamento vetorial, possibilitando a criação de aplicativos autênticos capazes de realizar ações para além de responder perguntas. Isso abre diversas oportunidades de uso prático, especialmente em assistência pessoal, abrangendo tarefas como

reservas de voos, transferências de dinheiro e auxílio em questões tributárias. Além disso, o LangChain tem implicações significativas no aprendizado de novos assuntos, permitindo que um modelo de linguagem referencie programas de estudo e acelere o processo de aprendizado. A integração de grandes modelos de linguagem a dados corporativos, como informações de clientes ou dados de marketing, por meio de APIs avançadas, como a API da Meta ou a API do Google, promete avanços exponenciais na análise de dados (BHATTI, 2023).

O LangChain oferece benefícios como a reutilização de modelos de linguagem específicos de domínio sem a necessidade de retreinamento, facilitando a criação de aplicações complexas que referenciam informações proprietárias para melhorar as respostas do modelo. Além disso, o LangChain simplifica o desenvolvimento de inteligência artificial ao abstrair a complexidade das integrações de fontes de dados e ao permitir a personalização de sequências para criar aplicações complexas rapidamente, reduzindo o tempo de desenvolvimento.

O framework oferece suporte aos desenvolvedores de IA ao fornecer ferramentas para conectar modelos de linguagem com fontes de dados externas, sendo de código aberto e contando com suporte de uma comunidade ativa. Com o LangChain, as organizações podem aproveitar suas funcionalidades gratuitamente e contar com o suporte de outros desenvolvedores proficientes na estrutura, facilitando o desenvolvimento de aplicações baseadas em modelos de linguagem.

O framework oferece suporte aos desenvolvedores de IA ao fornecer ferramentas para conectar modelos de linguagem com fontes de dados externas, sendo de código aberto e contando com suporte de uma comunidade ativa. Com o LangChain, as organizações podem aproveitar suas funcionalidades gratuitamente e contar com o suporte de outros desenvolvedores proficientes na estrutura, facilitando o desenvolvimento de aplicações baseadas em modelos de linguagem (ERICKSON, 2023).

A função de recuperação no LangChain desempenha um papel vital nas aplicações que necessitam de dados específicos do usuário, não incluídos no conjunto de treinamento do modelo em LangChain. Esse processo, denominado Geração com Recuperação Aprimorada (RAG), envolve a busca de dados externos e sua incorporação ao processo de geração do modelo de linguagem. Para facilitar esse procedimento em aplicações simples e complexas,

LangChain oferece um conjunto abrangente de ferramentas e funcionalidades. A recuperação em LangChain é realizada por meio de uma série de componentes, os quais serão discutidos individualmente.

Os carregadores de documentos no LangChain possibilitam a extração de dados de diversas fontes. Com uma seleção de mais de 100 carregadores disponíveis, eles oferecem suporte a uma variedade de tipos de documentos, aplicativos e fontes, como buckets S3 privados, sites públicos e bancos de dados. Você pode selecionar o carregador de documentos mais adequado às suas necessidades. Todos esses carregadores processam os dados na forma de aulas ISO.

O LangChain é uma biblioteca de abstrações desenvolvida para Python e Javascript, que representa as etapas e conceitos comuns essenciais para lidar com modelos de linguagem. Seus componentes modulares, como funções e classes de objetos, atuam como blocos de construção para programas de IA generativos. Esses componentes podem ser "encadeados" para criar aplicativos, reduzindo a quantidade de código e a compreensão refinada necessários para executar tarefas complexas de Processamento de Linguagem Natural (PLN). Embora a abordagem abstrata da LangChain possa impor limitações na personalização refinada para programadores especializados, ela proporciona a especialistas e iniciantes a capacidade de experimentar e prototipar rapidamente (IBM, 2023).

O LangChain é base para uma ferramenta desenvolvida pelo Tribunal de Contas do Estado de São Paulo (TCE-SP). O artigo desenvolvido por Xavier (2023) descreve a ANIA, uma Assistente Natural com Inteligência Artificial, desenvolvido pelo Tribunal de Contas do Estado de São Paulo (TCE-SP). A ANIA, que utiliza a mesma tecnologia do ChatGPT, é projetada para simplificar e acelerar a análise de documentos em PDF através de perguntas em linguagem natural e possui uma interface em português do Brasil. Ela oferece aos servidores do TCESP a capacidade de enviar documentos em PDF para análise, respondendo perguntas específicas, fornecendo resumos e estruturando os documentos em tópicos relevantes, mantendo a segurança e a privacidade das informações.

Esta nova ferramenta permite que os servidores do TCESP enviem documentos em PDF diretamente para a assistente, que, graças ao seu avançado poder de processamento, é capaz de analisar o texto contido nestes documentos, responder a perguntas específicas sobre o conteúdo, fornecer resumos concisos e até mesmo estruturar o documento em tópicos

relevantes. Tudo isso é feito com uma ênfase particular na segurança e na privacidade das informações, garantindo que os dados permaneçam protegidos dentro do ambiente controlado do TCE-SP.

Além de sua capacidade de processamento avançado e de manutenção da segurança da informação, a ANIA se destaca por várias razões. Primeiramente, sua tecnologia baseia-se no ChatGPT, reconhecido por sua confiabilidade e eficácia em uma ampla gama de aplicações. Isso fornece ao TCE-SP uma base sólida para a implementação de sua assistente de inteligência artificial.

Em segundo lugar, a ANIA foi cuidadosamente projetada para atender às necessidades específicas do tribunal, assegurando uma integração perfeita com o ambiente do TCESP e eliminando preocupações com privacidade e conformidade. Além disso, sua interface de usuário é notavelmente simples e intuitiva, o que facilita a interação com a assistente e permite que todos os colaboradores aproveitem os recursos avançados de inteligência artificial sem uma curva de aprendizado íngreme.

Um dos aspectos mais importantes da ANIA é sua capacidade de aprender continuamente com a interação com os usuários e o uso constante. Isso significa que a assistente se adapta progressivamente às necessidades específicas do TCE-SP, fornecendo respostas cada vez mais precisas e otimizando os fluxos de trabalho ao longo do tempo. Essa característica de aprendizado contínuo assegura que a assistente se torne uma ferramenta cada vez mais eficaz e personalizada para o tribunal.

## 3 Metodologia

Este capítulo descreve a abordagem metodológica adotada na pesquisa. Para alcançar os objetivos propostos, definiram-se três fases principais:

- Primeira Fase: Identificação do problema de pesquisa, delimitação do escopo do estudo e formulação dos objetivos gerais e específicos.
- Segunda Fase: Desenvolvimento de um chatbot utilizando a biblioteca LangChain para conduzir experimentações práticas com técnicas de engenharia de prompt.
- Terceira Fase: Execução de testes baseados nas experiências anteriores do TCMRio, aplicando técnicas de engenharia de prompt para aprimorar os resultados.

### 3.1 Definição do Problema

O Tribunal de Contas do Município do Rio de Janeiro (TCMRio), assim como outras instituições públicas, busca incorporar a inteligência artificial para otimizar seus processos. Com o avanço das IAs generativas, cenários como a atuação de auditores fiscais podem ser significativamente beneficiados pelo uso dessas tecnologias. O TCMRio utiliza atualmente a ferramenta Neural, fornecida pela Brasoftware, para um projeto-piloto. Este trabalho explora o uso de técnicas de engenharia de prompt para melhorar os resultados obtidos com ferramentas de IA generativa, desenvolvendo um chatbot com o framework LangChain para testes e refinamento de prompts.

### 3.2 Desenvolvimento do Chat

O chat foi criado usando o framework LangChain, empregado também em casos de uso como pelo TCE-SP. O projeto optou pela versão em TypeScript, dada a familiaridade prévia com essa linguagem. Utilizou-se o banco de dados Redis, conforme sugerido na documentação do LangChain, implementando uma aplicação backend que executa rotinas e finaliza, suficiente para os testes propostos.

### 3.2.1 Framework LangChain

A escolha do LangChain se deu pela facilidade em manipular componentes cruciais ao desenvolvimento de chats integrados com modelos de linguagem de grande escala (LLMs). O framework permite fácil integração com diversas APIs de IA generativa, promovendo flexibilidade e eficiência no desenvolvimento de soluções personalizadas. Além disso, o LangChain suporta a criação de fluxos de trabalho interativos, oferecendo uma interface intuitiva para a construção de cadeias de processamento de linguagem natural.

### 3.2.2 Banco Redis

O Redis foi utilizado por sua capacidade específica de trabalhar com vetores (Redis Vector Store), otimizando consultas. Esta escolha está alinhada com a documentação do LangChain. O código da figura 2 ilustra a criação de uma conexão com o banco, a instanciação do RedisVectorStore para o armazenamento de arquivos em vetores, e o OpenAIEmbeddings, responsável por transformar textos em vetores.

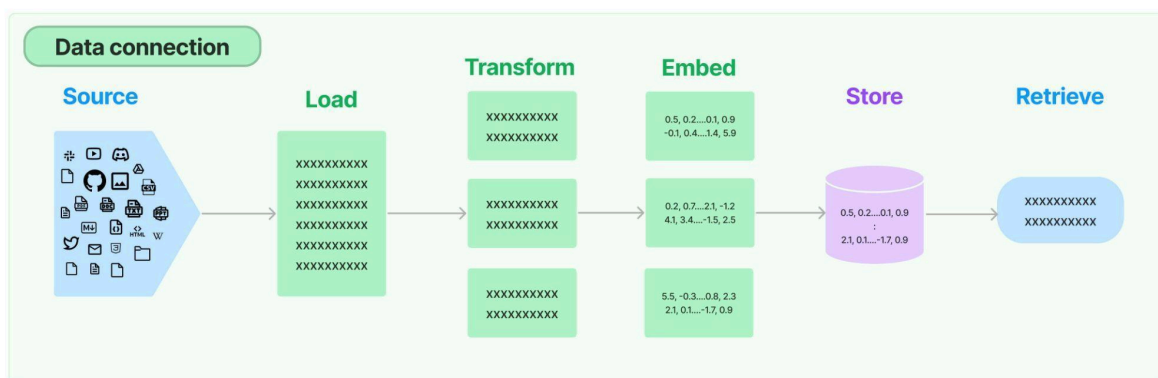
2.Figura código RedisVectorStore

```
1 import { RedisVectorStore } from 'langchain/vectorstores/redis';
2 import { OpenAIEmbeddings } from 'langchain/embeddings/openai';
3 import { createClient } from 'redis';
4
5 const OPENAI_API_KEY = 'aaaaabbbbccccccddddeeeeffffffggggg';
6
7 export const redis = createClient({
8   url: 'redis://127.0.0.1:6379'
9 });
10
11 export const redisVectorStore = new RedisVectorStore(
12   new OpenAIEmbeddings({ openAIApiKey: OPENAI_API_KEY }),
13   {
14     indexName: 'arquivos-teste-embeddings',
15     redisClient: redis,
16     keyPrefix: 'arquivos-teste:'
17   }
18 );
```

2. Fonte: Elaborada pelo autor

Na figura 3 é possível observar o fluxo que os dados da base privada do usuário fazem até que estejam prontos para serem utilizados para busca de similaridade. Os dados textuais, como PDFs ou outros formatos, são recebidos pela aplicação, divididos em chunks, transformados em vetores através do processo de embedding, e armazenados no banco.

3. Figura de fluxo de embedding dos dados para serem armazenados



3. Fonte: Documentação do framework LangChain

### 3.2.3 Api OpenAI

A integração com a API da OpenAI foi uma escolha estratégica pela qualidade dos modelos de linguagem, como o GPT, e pela documentação detalhada e atualizações constantes, facilitando a integração e manutenção do projeto atualizado com as inovações em IA.

### 3.2.4 Prompt

Na figura 4 é demonstrado como o prompt está estruturado. Ele é dividido em 3 partes principais: o prompt estático, a pergunta do usuário e a resposta da busca de similaridade no banco.

4. Estrutura do prompt

```

14 const prompt = new PromptTemplate({
15   template: `
16     Você responde perguntas sobre jurisprudência, acórdão, súmula entre outros termos relacionados
17     a práticas de um tribunal de contas e auditores fiscais;
18     O usuário está buscando ajuda para entender melhor sobre coisas relacionadas
19     a um tribunal de contas e suas atividades;
20     Use o conteúdo dos textos abaixo para responder a pergunta do usuário,
21     ajude-o pois ele pode ser um leigo, um auditor fiscal, um conselheiro, um juiz, etc.;
22     Se a resposta não for encontrada nos textos, responda que você não sabe, não tente inventar
23     uma resposta;
24
25     Se possível, inclua referências dos textos abaixo que auxiliaram a resposta.
26
27     Textos:
28     {context}
29
30     Pergunta:
31     {question}
32   `.trim(),
33   inputVariables: ['context', 'question'],
34 });

```

4. Fonte: Elaborada pelo autor

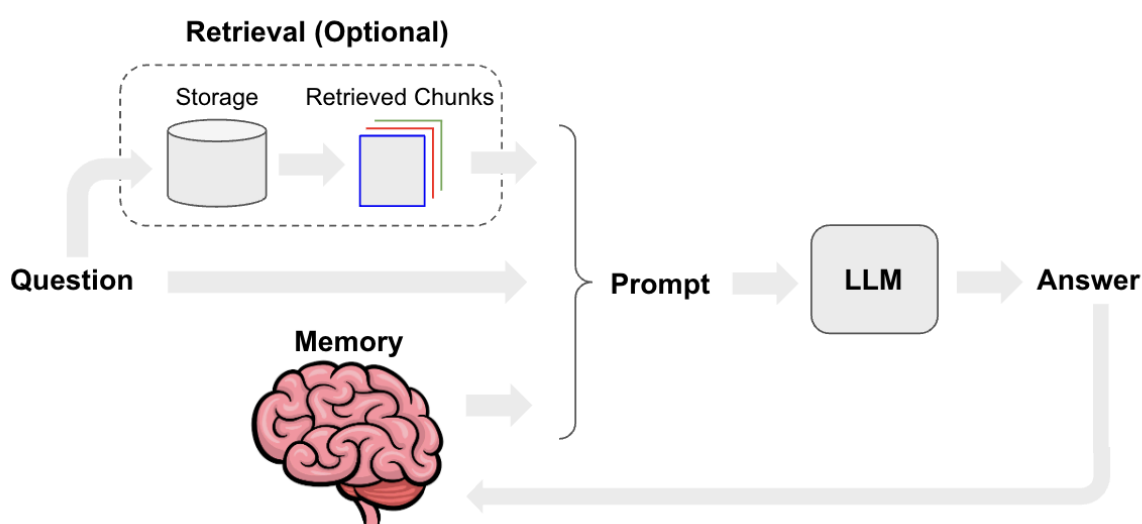
O prompt estático é o lugar onde é passado instruções de contexto para embasar melhor as respostas do LLM. Já nele é possível notar o uso de pelo menos 4 técnicas de engenharia de prompt:

- **Técnica de Role-Playing (Role-Playing Prompting):** Esta abordagem envolve configurar cenários em que a IA adota uma persona ou papel específico, tal como um especialista em um campo particular ou um personagem de ficção. O objetivo é enriquecer as respostas com perspectivas únicas ou abordagens especializadas que se alinham ao papel assumido.
- **Técnica de Contextualização (Contextualization Prompting):** Consiste em incorporar ao prompt detalhes contextuais significativos, facilitando o entendimento do modelo sobre o cenário ou tema abordado. Isso ajuda a IA a gerar respostas mais precisas e contextualmente adequadas.
- **Prompts Negativos e Positivos (Negative/Positive Prompting):** Esta técnica envolve instruções claras sobre o que deve ser evitado ou enfatizado nas respostas. Por meio da inclusão de diretrizes específicas, é possível moldar o conteúdo gerado para que este se alinhe mais estreitamente com os requisitos ou expectativas desejadas.

- **Prompting Híbrido (Hybrid Prompting):** A combinação de múltiplas estratégias de prompt pode ser utilizada para otimizar os resultados. Ao integrar diferentes técnicas, como role-playing, contextualização, e direcionamentos negativos ou positivos, busca-se uma abordagem mais robusta e flexível que potencialize a eficácia das interações com o modelo de IA.

A pergunta feita pelo usuário também é uma parte fundamental. O usuário final não tem acesso a configuração de prompt estático, mas ao utilizar as técnicas de engenharia de prompt, ele aumenta as chances de que sua pergunta traga melhor resultado. A figura 5 mostra como é o fluxo completo desde a pergunta do usuário até a resposta final depois de passar pelo LLM:

5. Fluxo de chat implementado com LangChain



5. Fonte: Elaborada pelo autor

A pergunta do usuário primeiro é transformada em vetor e usada para fazer a busca de similaridade na base privada (banco redis onde os textos que passaram pelo processo de embedding estão salvos). Como resposta desta consulta na base privada vem os chunks, eles são adicionados ao Prompt Template. A pergunta do usuário também é adicionada para completar o Prompt Template que será enviado na requisição à api integrada ao chat.

A figura 6 mostra como foi configurado, no caso dos testes, para retornar 3 chunks de 600 tokens cada.

## 6. Configuração dos chunks

```
const splitter = new TokenTextSplitter({
  encodingName: 'cl100k_base',
  chunkSize: 600,
  chunkOverlap: 0,
});

const chain = RetrievalQAChain.fromLLM(openAiChat, redisVectorStore.asRetriever(3), {
  prompt,
  returnSourceDocuments: true,
  verbose: true,
});
```

6. Fonte: Elaborada pelo autor

## 4. Resultados

Este estudo explorou a implementação de técnicas de engenharia de prompt na análise de documentos pelo Tribunal de Contas do Município do Rio de Janeiro (TCMRio), aproveitando os modelos de linguagem de grande escala (LLM) fornecidos pela OpenAI. Adotou-se uma metodologia comparativa, analisando diferentes iterações dos modelos GPT e a eficiência de uma ferramenta neural desenvolvida pela Brasoftware. O objetivo era avaliar a viabilidade dessas tecnologias como ferramentas de apoio aos funcionários do tribunal. Para isso, foram realizados testes de engenharia de prompt, com a preparação e aprimoramento de perguntas por meio dessas técnicas. Além disso, foi desenvolvido um chat utilizando o framework LangChain para realizar testes práticos.

### 4.1 Implementação de Testes

Os testes iniciais foram com a versão gpt-3.5-turbo, após evoluir à integração para a versão gpt-4.5-turbo, observou-se uma melhoria no detalhamento das respostas, mantendo, contudo, o mesmo percentual de aproveitamento. Este fenômeno sugere que, apesar da evolução do modelo de IA, o mecanismo de consulta à base de conhecimento e as restrições impostas à formulação de prompts influenciaram diretamente a performance obtida.

Para realizar os testes foi utilizado como base os testes feitos pelo time do TCMRio. No presente capítulo estão tanto as perguntas como as respostas obtidas na Prova de Conceito do tribunal. Essas perguntas foram alteradas com base em técnicas de engenharia de prompt, para que se obtivesse melhor aproveitamento nas respostas.

Tabela 1. Perguntas originais da Poc TCMRio e com engenharia de prompt.

<b>PERGUNTAS ORIGINAIS Poc TCMRio</b>	<b>PERGUNTAS MELHORADAS Utilização de Engenharia de Prompt</b>
1- Quantas súmulas o TCMRio possui atualmente?	1- "Considerando a importância de manter informações atualizadas para análise e referência, poderia informar qual é o número exato de súmulas publicadas pelo Tribunal de Contas do Município do Rio de Janeiro (TCMRio) até o presente momento?"

	Essa informação é crucial para garantir a precisão no estudo e aplicação do direito administrativo relacionado ao município."
2- Quantas súmulas o TCMRJ possui atualmente?	2- "Em um esforço para compreender a amplitude das diretrizes legais estabelecidas pelo Tribunal de Contas do Município do Rio de Janeiro (TCMRJ), poderia fornecer o total de súmulas oficialmente adotadas por este tribunal até a data atual? Esta informação será utilizada para uma análise detalhada do impacto dessas súmulas nas decisões administrativas municipais."
3- Me relacione 3 súmulas do TCMRJ.	3- "Dado o papel significativo que as súmulas do TCMRJ desempenham em orientar a jurisprudência administrativa, poderia selecionar e descrever três súmulas emitidas por este tribunal? Inclua, por favor, os números e um breve resumo de cada súmula, enfatizando suas implicações práticas para a administração municipal."
4- Licença para tratamento da própria saúde ou licença por motivo de doença em pessoa da família são como tempo.	4- "Relativo às políticas de gestão de pessoal no âmbito municipal, como o TCMRJ interpreta o período de licença para tratamento da própria saúde ou por motivo de doença em pessoa da família em termos de contagem de tempo de serviço? Por favor, explique o raciocínio jurídico que fundamenta essa interpretação e suas implicações para os funcionários municipais."
5- Esse entendimento é uma súmula do TCMRJ?	5- "Referente à compreensão de normas administrativas, este entendimento sobre a contabilização de licenças médicas como tempo de serviço é consolidado em alguma súmula do

	TCMRJ? Se sim, por favor, forneça uma análise crítica sobre como esta súmula influencia a gestão de recursos humanos no setor público municipal."
6- Quais são as súmulas do Tribunal de Contas do Município do Rio de Janeiro?	6- "Com o objetivo de aprofundar o conhecimento sobre as normativas que regem o funcionamento administrativo municipal, poderia listar e explicar os temas abordados nas súmulas vigentes do Tribunal de Contas do Município do Rio de Janeiro? Essa exploração visa identificar áreas de potencial inovação na aplicação do direito administrativo."
7- Bom dia! De acordo com a jurisprudência do TCMRio, o tempo de serviço como residente médico pode ser utilizado para fins de aposentadoria?	7- "Assumindo o papel de um consultor jurídico especializado em direito administrativo, poderia informar se o tempo de serviço realizado como residente médico é reconhecido para fins de aposentadoria pelo TCMRio? Além disso, destaque a súmula ou decisão que aborda esse tema, detalhando seu impacto nas políticas de gestão de pessoal."
8- Perguntou-se então: Este entendimento foi objeto de súmula do TCMRio?	8- "Baseado na discussão anterior sobre a contagem de tempo de serviço para residentes médicos, esse entendimento foi consolidado em alguma súmula específica pelo TCMRio? Se sim, descreva o processo lógico que levou à formação dessa súmula e como ela se aplica no contexto administrativo atual."
9- Você está certo disso? Do que se trata a súmula 11 do TCMRio?	9- "Imagine que você está preparando uma apresentação sobre as diretrizes administrativas mais influentes do TCMRio. Dentro deste cenário, como você descreveria o conteúdo e a importância da súmula número 11 para o público interessado em

	governança municipal? Inclua detalhes que realcem seu valor prático e teórico."
10- Qual é o enunciado da súmula nº 11 do TCMRio?	10- "Através de uma abordagem socrática, questione-se sobre o texto exato e as implicações da súmula número 11 emitida pelo TCMRio. Como este enunciado influencia as práticas administrativas municipais e quais são os desafios e oportunidades associados à sua implementação?"
11- Cite 3 súmulas do TCMRio.	11- "Utilizando a técnica de auto-indução, poderia autogerar uma lista de três súmulas do TCMRio, fornecendo um breve resumo e análise de cada uma? Esta tarefa visa explorar a aplicabilidade e o impacto dessas súmulas na administração pública municipal, destacando exemplos práticos de sua implementação."
12- O TCMRJ possui boletins de jurisprudência?	12- "Em um esforço para mapear o acesso à informação jurídica municipal, o TCMRJ publica boletins de jurisprudência de forma regular? Se sim, detalhe o processo para acessar esses boletins e como eles podem ser utilizados por profissionais do direito e administradores públicos para melhorar a governança municipal."
13- De acordo a jurisprudência do TCM, como a jurisdicionada deve agir na ocorrência de desistências em processos de pregão?	13- "Considerando os desafios enfrentados em processos de pregão, especialmente em casos de desistência, como o TCMRJ orienta as jurisdicionadas a proceder? Contrastando com práticas menos eficazes, destaque as recomendações do TCMRJ para garantir a integridade e eficiência desses processos."

<p>14- De acordo com a jurisprudência do TCM, existe alguma regra para aplicar critérios da dosimetria na aplicação de multas?</p>	<p>14- "Dentro do contexto de aplicação de sanções administrativas, como o TCMRJ estabelece critérios para a dosimetria de multas? Através de um exercício criativo, explore como esses critérios são determinados e aplicados, considerando o equilíbrio entre justiça e eficácia administrativa."</p>
<p>15- De acordo com a jurisprudência do TCMRio, o que deve ser observado com relação aos embargos de declaração?</p>	<p>15- "Na avaliação de embargos de declaração, quais critérios o TCMRio considera essenciais para sua análise? Utilizando um método socrático, examine as diretrizes estabelecidas pelo TCMRio para o tratamento desses embargos, avaliando sua importância para a clareza e a precisão jurídica nas decisões administrativas."</p>

1. Fonte: Elaborada pelo autor

Na tabela 2 está apresentado as respostas que foram devolvidas tanto na PoC do TCMRio, quanto nos testes feitos com o chat desenvolvido utilizando engenharia de prompt para melhor as respostas do LLM.

Tabela 2. Respostas originais da Poc TCMRio e respostas do Chat LangChain.

<p><b>RESPOSTAS Poc TCMRio</b></p>	<p><b>RESPOSTAS Chat LangChain</b></p>
<p>1- Não encontrei informações na base de conhecimento interna para responder a sua pergunta sobre o número de súmulas do temrio.</p>	<p>1- O Tribunal de Contas do Município do Rio de Janeiro (TCMRio) possui, com base nas informações fornecidas, pelo menos 11 súmulas. Isso pode ser inferido a partir da menção à "Súmula n.º 011" nos textos fornecidos. No entanto, é importante notar que pode haver mais súmulas além dessas, mas com base nos dados disponíveis, apenas 11 súmulas são explicitamente mencionadas.</p>

<p>2- Não encontrei informações na base de conhecimento interna para responder a sua pergunta sobre o número de súmulas do temrio.</p>	<p>2- Com base nos textos fornecidos, não é possível determinar o número total de súmulas que o Tribunal de Contas do Município do Rio de Janeiro (TCMRJ) possui atualmente. A referência feita nos textos menciona apenas até a Súmula nº 004, sem indicar se existem mais súmulas além dessa. Portanto, para obter informações atualizadas sobre o total de súmulas do TCMRJ, seria necessário consultar diretamente fontes oficiais do Tribunal.</p>
<p>3- Desculpe, mas não encontrei informações na base de conhecimento interna sobre as súmulas do TCMRJ.</p>	<p>3- As súmulas do Tribunal de Contas do Município do Rio de Janeiro (TCMRJ) são resumos de entendimentos consolidados que servem como referência para a tomada de decisões. No entanto, com base nos textos fornecidos, há menção direta a apenas uma súmula específica:       **Súmula nº 004**</p>
<p>4- Resposta positiva.</p>	<p>4- Licença para tratamento da própria saúde ou licença por motivo de doença em pessoa da família não são computadas como tempo de efetivo exercício no serviço público para fins de aposentadoria. Isso está estabelecido na Súmula n.º 013 do Tribunal de Contas do Município do Rio de Janeiro, que especifica que os períodos de afastamentos relativos à licença para tratamento da própria saúde (art. 88 da Lei 94/1979) e à licença por motivo de doença em pessoa da família (art. 100 da Lei 94/1979) não devem ser considerados como tempo de efetivo exercício no serviço público para propósitos de aposentadoria.</p>

<p>5- Desculpe, mas não encontrei informações específicas sobre as súmulas do TCMRJ.</p>	<p>5- Sim, você está correto. De acordo com a Súmula n.º 013 do Tribunal de Contas do Município do Rio de Janeiro (TCMRJ), os períodos de afastamento devido à licença para tratamento da própria saúde, conforme o artigo 88 da Lei 94/1979, e à licença por motivo de doença em pessoa da família, conforme o artigo 100 da mesma lei, não são computados como tempo de efetivo exercício no serviço público para fins de aposentadoria. Este entendimento é reforçado por precedentes citados na própria súmula, demonstrando a consistência da aplicação desta interpretação em diversas decisões do tribunal.</p>
<p>6- Desculpe, mas não encontrei informação específica sobre as súmulas do Tribunal de Contas do município do Rio de Janeiro na base de conhecimento interna.</p>	<p>6- Os textos fornecidos não listam súmulas do Tribunal de Contas do Município do Rio de Janeiro. Eles contêm resumos de decisões proferidas pelo tribunal em determinados períodos, mas não mencionam súmulas especificamente. As súmulas são enunciados que sintetizam entendimentos consolidados ou majoritários que orientam a atuação do tribunal, mas para obter informações específicas sobre súmulas do Tribunal de Contas do Município do Rio de Janeiro, seria necessário consultar diretamente as fontes oficiais do tribunal.</p>
<p>7- OBS: A resposta não citou a súmula.</p>	<p>7- Bom dia! Sim, de acordo com a jurisprudência do Tribunal de Contas do Município do Rio de Janeiro (TCMRJ), o tempo de serviço apurado como residente médico pode ser utilizado para fins de aposentadoria. Especificamente, a Súmula n.º 011 estabelece que o aproveitamento do tempo de serviço como residente médico até 1º/11/1982 dispensa a prova de recolhimento de contribuição previdenciária como segurado autônomo para fins de aposentadoria, mesmo para aqueles servidores que não tenham reunido as condições de aposentação até 27/4/2011.</p>

	<p>Além disso, a Súmula n.º 010 admite, excepcionalmente, para os servidores que tenham reunido os requisitos de aposentação até 27/4/2011, a averbação do tempo de serviço prestado como estagiário, acadêmico bolsista ou residente médico, para fins de aposentadoria, independentemente da comprovação de recolhimento de contribuição previdenciária como segurado autônomo.',</p>
<p>8- Desculpe, mas não encontrei informação específica sobre as súmulas do Tribunal de Contas do município do Rio de Janeiro na base de conhecimento interna.</p>	<p>8- Sim, este entendimento foi objeto de súmula do Tribunal de Contas do Município do Rio de Janeiro (TCMRio). Conforme as Súmulas n.º 011 e n.º 010 do TCMRio, há disposições específicas sobre o aproveitamento do tempo de serviço para fins de aposentadoria, incluindo o tempo de serviço como residente médico. A Súmula n.º 011 especifica que o tempo de serviço apurado como residente médico até 1º/11/1982 dispensa a prova de recolhimento de contribuição previdenciária como segurado autônomo para fins de aposentadoria, mesmo para os servidores que não tenham reunido as condições de aposentação até 27/4/2011. Já a Súmula n.º 010 permite, de forma excepcional, para os servidores que reuniram os requisitos de aposentação até 27/4/2011, a averbação do tempo de serviço prestado em condições específicas, como residente médico, para fins de aposentadoria, sem a necessidade de comprovação de recolhimento de contribuição previdenciária como segurado autônomo.'</p>
<p>9- Resposta alucinada:</p>	<p>9- A Súmula n.º 011 do Tribunal de Contas do Município do Rio de Janeiro trata da possibilidade de aproveitamento do tempo de serviço apurado como residente médico até 1º/11/1982 para fins de aposentadoria, sem a necessidade de comprovação de recolhimento de contribuição previdenciária como segurado autônomo. Isso se aplica mesmo para</p>

	<p>aqueles servidores que não tenham reunido as condições necessárias para a aposentadoria até 27/4/2011. Esta súmula reflete uma decisão baseada em precedentes judiciais que reconhecem essa possibilidade, facilitando assim o processo de aposentadoria para os servidores que se enquadram nesses critérios.</p>
<p>10- Não foi preciso na resposta, não citou como referência o arquivo da súmula 11. Não encontrou o enunciado da súmula nº 11, mesmo perguntando de forma direta.</p>	<p>10- Bom dia! Sim, de acordo com a jurisprudência do Tribunal de Contas do Município do Rio de Janeiro (TCMRio), o tempo de serviço apurado como residente médico pode ser utilizado para fins de aposentadoria. Especificamente, a Súmula n.º 011 do TCMRio estabelece que o aproveitamento do tempo de serviço como residente médico até 1º/11/1982 dispensa a prova de recolhimento de contribuição previdenciária como segurado autônomo para fins de aposentadoria, mesmo para aqueles servidores que não tenham reunido as condições de aposentação até 27/4/2011. Além disso, a Súmula n.º 010 admite, excepcionalmente, para os servidores que tenham reunido os requisitos de aposentação até 27/4/2011, a averbação do tempo de serviço prestado como estagiário, acadêmico bolsista ou residente médico, para fins de aposentadoria, independentemente da comprovação de recolhimento de contribuição previdenciária como segurado autônomo.</p>
<p>11- Alucinação Conclusão: O Neural continua a não entender o conteúdo de súmulas do TCMRio</p>	<p>11- Aqui estão três súmulas do Tribunal de Contas do Município do Rio de Janeiro (TCMRio):\n' +  'n' +  '1. <b>Súmula nº 004</b>: "Após o registro inicial de aposentadorias e pensões, somente devem ser remetidos ao TCMRJ processos nos quais tenha ocorrido mudança de fundamento legal do ato de</p>

	<p>concessão ou das parcelas remuneratórias." (Fundamento Legal - Artigo 71, III, da Constituição Federal).\n' +</p> <p>\n' +</p> <p>2. <b>Súmula n.º 007</b>: "Nas licitações, a apuração dos valores estimados será estabelecida tomando por base o menor preço e, nos casos em que a legislação permita a sua não utilização, deverá o órgão justificar o critério escolhido desconsiderando as propostas incompatíveis com a realidade de mercado, em atenção ao Princípio da Economicidade." (Fundamento Legal - Lei n.º 8666/93, artigos 3.º e 45, §1º, inciso I).\n' +</p> <p>\n' +</p> <p>3. <b>Súmula n.º 009</b>: "Viola o princípio da economicidade a não utilização da média aritmética da distância entre o local da obra e os 3 (três) locais de descarte de materiais de obras e serviços de engenharia mais próximos, considerando os existentes na lista das empresas licenciadas para a destinação ambiental de Resíduos de Construção Civil, para fins de estimativa da Distância Média de Transporte (DMT) usada na formulação do preço de referência, ressalvada a possibilidade de haver justificação técnica no processo que demonstre ser mais vantajosa a adoção de outro critério.",</p>
<p>12- Respostas incorretas e imprecisas. Considerei uma alucinação</p>	<p>12- Sim, o Tribunal de Contas do Município do Rio de Janeiro (TCMRJ) possui boletins de jurisprudência. Esses boletins consolidam, de forma sintética, as decisões proferidas pelo Colegiado do TCMRJ consideradas relevantes sob o aspecto jurisprudencial. O informativo mencionado cobre as sessões do Plenário e das Câmaras de Janeiro a Julho de 2019, oferecendo um resumo das decisões e</p>

	<p>permitindo o acesso ao inteiro teor do voto por meio de links disponíveis. Este mecanismo facilita a compreensão e o estudo da jurisprudência do tribunal para todos os interessados, sejam eles leigos ou profissionais do direito, como auditores fiscais, conselheiros ou juizes.</p>
<p>13- Resposta imprecisa, confusa e não traz de forma clara como a jurisdicionada deve agir. Essa resposta consta em um enunciado no boletim jurisprudencial nº 06</p>	<p>13- De acordo com a jurisprudência do Tribunal de Contas do Município (TCM), na ocorrência de desistências tácitas em processos de pregão, a jurisdicionada deve instaurar um processo administrativo para apurar eventual penalidade em relação às empresas desistentes. Isso se aplica caso não haja justificativa plausível para tais desistências. A análise conjunta de ata de registro de preços e edital de pregão eletrônico pelo TCM considerou que a não apresentação de documentação pelo licitante, que ofereceu lances mas não se manifestou nas etapas subsequentes do pregão, configura desistência tácita da participação no certame. Assim, foi determinado que, na eventualidade de não haver justificativas aceitáveis para a omissão, cabe ao órgão licitante julgar se a argumentação merece ser acolhida ou não e, conforme o caso, proceder com a instauração do processo administrativo para aplicação das penalidades previstas tanto no edital quanto no art. 7º da Lei n.º 10.520/2002 em relação às empresas licitantes que foram desclassificadas (Boletim Jurisprudencial, Sessões do Plenário de Janeiro a Junho de 2020).'</p>
<p>14- A resposta abordou o tema, mas não citou o enunciado, mesmo perguntando diretamente</p>	<p>14- Sim, de acordo com a jurisprudência do Tribunal de Contas Municipal (TCM), existe uma regra específica para a aplicação de critérios de dosimetria na aplicação de multas. Conforme o caso analisado no processo 40/1613/2013, relatado pelo Conselheiro Felipe Galvão Puccioni, é válida a utilização dos</p>

	<p>critérios de dosimetria previstos pelo direito penal quanto à fixação da pena-base na aplicação de sanção de multa pelo TCM. O relator indicou que é lícito traçar um paralelo entre o procedimento de aplicação da pena de multa pelo Tribunal de Contas e a primeira fase do procedimento de aplicação das sanções penais, referindo-se à fase de fixação da pena-base.</p>
<p>15- Desculpe, mas não encontrei informações específicas sobre o assunto.</p>	<p>15- Sim, de acordo com a jurisprudência do Tribunal de Contas Municipal (TCM), existe uma regra específica para a aplicação de critérios de dosimetria na aplicação de multas. Conforme o caso analisado no processo 40/1613/2013, relatado pelo Conselheiro Felipe Galvão Puccioni, é válida a utilização dos critérios de dosimetria previstos pelo direito penal quanto à fixação da pena-base na aplicação de sanção de multa pelo TCM. O relator indicou que é lícito traçar um paralelo entre o procedimento de aplicação da pena de multa pelo Tribunal de Contas e a primeira fase do procedimento de aplicação das sanções penais, referindo-se à fase de fixação da pena-base</p>

2. Fonte: Elaborada pelo autor

Esses testes foram realizados com apenas 15 perguntas retiradas do material de testes fornecido pelo Tribunal de Contas do Município do Rio de Janeiro, vale ressaltar que este é apenas um recorte dos dados, pois o material fornecido é muito maior. Também é importante dizer que o Chat desenvolvido para auxiliar estes testes, teve acesso ao mesmo conjunto de documentos que a ferramenta Neural da Brasoftware. Tais documentos foram armazenados localmente para servir de consulta como base privada, seguindo o modelo descrito em tópicos anteriores.

A análise dos resultados mostra que de 15 questões avaliadas apenas 1 teve uma resposta positiva e nas outras 14, a PoC do TCMRio, não devolveu respostas satisfatórias. Resultando em apenas 6,6% de aproveitamento. Em contrapartida, as perguntas depois de reformuladas com uso de engenharia de prompt forneceram 11 respostas positivas 4 incompletas atingindo 72,6% de aproveitamento.

Isso evidencia a importância crucial das técnicas de engenharia de prompt na otimização da interação com modelos de linguagem artificial para fins de análise. Foi possível constatar que a precisão na formulação dos prompts e a estratégia de consulta à base de conhecimento privada são determinantes para a qualidade e relevância das informações obtidas por meio dessas ferramentas.

## **4.2 Desafios Identificados**

- **Consulta à Base de Conhecimento Privada:** A dependência dos modelos em relação aos dados disponíveis na base de conhecimento privada ressalta a necessidade de uma curadoria cuidadosa e de um enriquecimento constante desta base para suportar respostas informadas.
- **Restrições de Prompt Estático:** A observação de que a configuração de prompts estáticos pode limitar a capacidade dos modelos de gerar respostas inovadoras e criativas destaca o equilíbrio necessário entre precisão e flexibilidade na interação com sistemas de IA.

## **4.3 Conclusões e Implicações para Futuras Pesquisas**

Este estudo destaca a importância crucial da engenharia de prompts para otimizar o desempenho de modelos de linguagem de grande escala (LLM) em contextos jurídicos, exemplificado pela análise de documentos do Tribunal de Contas do Município do Rio de Janeiro (TCMRio). Revelou-se que, apesar dos avanços trazidos pelas versões mais recentes do GPT, ainda enfrentamos desafios significativos, especialmente no que se refere ao acesso a dados relevantes e à criação de prompts que produzam respostas precisas e diretamente aplicáveis.

A pesquisa enfatiza a necessidade de um aprimoramento contínuo nas bases de conhecimento específicas e avalia a eficácia de técnicas de engenharia de prompts. Além disso, recomenda-se a execução de estudos futuros que investiguem como diferentes configurações de modelos de IA podem influenciar a precisão e a utilidade das respostas obtidas.

A dependência de sistemas de IA em bases de conhecimento especializadas destaca um desafio essencial: a capacidade de formular perguntas por meio da engenharia de prompts se torna um elemento chave para assegurar pesquisas eficientes e evitar respostas genéricas ou imprecisas. Tal estratégia sublinha a importância de prevenir "alucinações" dos modelos de IA, que podem afetar a confiabilidade das informações fornecidas, mesmo aquelas originárias de fontes externas confiáveis. A elaboração cuidadosa de prompts não somente eleva a relevância e a precisão das respostas, mas também garante que as consultas à base de conhecimento privado sejam realizadas com máxima eficiência e eficácia.

## Referências

BHATTI S. (2023) Tutorial LangChain: Libere o poder dos modelos de linguagem para tarefas versáteis! Disponível em: <https://hashdork.com/pt/langchain-tutorial/> Acesso em: 29 dez 2023

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; Amodei, D. Language Models are Few-Shot Learners. 2020. Disponível em: <https://arxiv.org/abs/2005.14165>. Acesso em: 29 dez 2023

ERICKSON, (2023) Busca com IA com a privacidade em primeiro lugar e usando o LangChain e o Elasticsearch

Disponível em:

<https://www.elastic.co/pt/blog/privacy-first-ai-search-langchain-elasticsearch> Acesso em: 20 jan 2024

Ferreira, E. Dias. Engenheiro de Prompt: Nova Profissão em Ascensão. 2023. Disponível em: <https://ica.ele.puc-rio.br/blog/engenheiro-de-prompt-nova-profissao-em-ascensao/>. Acesso em: 20 jan 2024

Fabio G. Cozman; Guilherme Ary Plonski; Hugo Neri. Inteligência Artificial: Avanços e Tendências (2021, p.22)

GADELHA, J. (2023) Dominando o Prompt Engineering: Boas Práticas e Técnicas para Maximizar o Potencial do GPT Disponível em: <https://medium.com/@joaoricardotg/dominando-o-prompt-engineering-boas-pr%C3%A1ticas-e-t%C3%A9cnicas-para-maximizar-o-potencial-do-gpt-3-102fdfe3b6ea> Acesso em: 29 jan 2024

GONÇALVES, T. (2023) PLN: o que é Processamento de Linguagem Natural? Disponível em: <https://www.alura.com.br/artigos/o-que-e-pln> Acesso em: 29 dez 2023

IARA, (2023) 5 Técnicas de Engenharia de Prompt e Como Elas Podem Melhorar a Sua Experiência com a IAs Generativas Disponível em: <https://iara.chat/5-tecnicas-de-engenharia-de-prompt-e-como-elas-podem/> Acesso em: 29 dez 2023

KLEINA, N. (2023) O que é GPT-4 Turbo, a versão mais poderosa do ChatGPT Disponível em: <https://www.mundoconectado.com.br/inteligencia-artificial/o-que-e-gpt-4-turbo-a-versao-mais-poderosa-do-chatgpt/> Acesso em: 26 dez 2023

KOVACS, L. (2023) O que é processamento de linguagem natural? [PLN] Disponível em: <https://tecnoblog.net/responde/o-que-e-processamento-de-linguagem-natural-nlp/> Acesso em: 29 dez 2023

IBM (2023) What is natural language processing? Disponível em: <https://www.ibm.com/br-pt/topics/natural-language-processing> Acesso em: 29 dez 2023

MAGALHÃES, A. L. (2023) O que é LLM? | Large Language Model Disponível em: <https://canaltech.com.br/inteligencia-artificial/o-que-e-llm-large-language-model/> Acesso em: 26 dez 2023

ORACLE. O que é IA? Saiba mais sobre inteligência artificial. [S. l.], 2023. Disponível em: <https://www.oracle.com/br/artificial-intelligence/what-is-ai/>. Acesso em: 28 dez. 2023.

REIS, R. (2023) LangChain — Guia de Início Rápido Disponível em: <https://ricardo-reis.medium.com/langchain-guia-de-in%C3%ADcio-r%C3%A1pido-138284ec8681> Acesso em: 26 dez 2023

Russell, S. J.; Norvig, P. Inteligência Artificial. 3ª ed. São Paulo: Pearson, 2013.

STRAZZI, A. (2023) Desvendando o ChatGPT na Advocacia: 3 Estratégias Éticas para Aumentar a Produtividade Disponível em:

<https://www.jusbrasil.com.br/artigos/desvendando-o-chatgpt-na-advocacia-3-estrategias-eticas-para-aumentar-a-produtividade/1991388336> Acesso em: 29 jan 2024

TOPSAKAL, O.; AKINCI, T. C. Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. *Computer Science Department, Florida Polytechnic University, FL, USA; WCGEC, University of California at Riverside, CA, USA, 2020*

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). Disponível em: <https://papers.nips.cc/paper/7181-attention-is-all-you-need>. Acesso em: 29 jan 2023

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. Disponível em: <https://ar5iv.org/abs/2302.11382>. Acesso em: 29 jan 2024