UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Metodologia Orientada a Agentes de Linguagem para Assistência Automotiva: Integrando Engenharia de *Prompts* em *Chatbots* Avançados

Thaís de Araújo de Medeiros

Orientador: Prof. Dr. Ivanovitch Medeiros Dantas da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem do PPgEEC: M692
Natal, RN, julho de 2025

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

Medeiros, Thaís de Araújo de.
Metodologia orientada a agentes de linguagem para assistência automotiva: integrando engenharia de Prompts em Chatbots avançados / Thaís de Araújo de Medeiros. - 2025.
81 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, 2025.
Orientação: Prof. Dr. Ivanovitch Medeiros Dantas da Silva.

1. Retrieval-Augmented Generation - Dissertação. 2. Assistência automotiva - Dissertação. 3. Engenharia de Prompts - Dissertação. 4. Modelos de linguagem - Dissertação. 5. Self-RAG - Dissertação. 6. Gradiente descendente - Dissertação. I. Silva, Ivanovitch Medeiros Dantas da. II. Título.

RN/UF/BCZM CDU 629

Elaborado por Jackeline dos Santos Pinheiro da Silva Maia Cavalcanti - CRB-15/317

Aos meus pais, Cleide e Hermano, que são a base de todas as minhas conquistas, por toda a dedicação e esforço.

Agradecimentos

Aos meus pais, Hermano José de Medeiros e Maria Cleide de Araújo Medeiros, por toda a luta para que eu pudesse ter uma educação de qualidade.

À minha avó, Maria de Lourdes Medeiros, pelo apoio de sempre.

Às minhas primas, Maria Eduarda Medeiros Monteiro e Ananda de Medeiros Monteiro, por toda a ajuda e companheirismo.

À minha tia, Edilene Garcia de Medeiros, pelo acolhimento em sua casa durante o início da minha vida universitária.

À madrinha Ana Paula, que fez questão de revisar cada página deste trabalho.

Ao meu orientador e professor, Ivanovitch Medeiros Dantas da Silva, por toda a paciência e suporte ao longo do mestrado. Agradeço por ser um professor que se dedica a seus orientandos, sempre enxerga além do óbvio e nos estimula a alcançar nosso melhor.

À professora Marianne Batista Diniz da Silva, por todos os conselhos e por ter dedicado seu tempo para revisar este trabalho, mesmo sem um vínculo formal de coorientação.

Aos meus amigos e colegas do Conect2ai, pela colaboração e pelos momentos de leveza que enriqueceram minha jornada acadêmica. Em especial, a Morsinaldo Medeiros, pela amizade e parceria desde quase o início da graduação; a Matheus Diniz e Miguel Euripedes, pelo companheirismo diário, que sempre combinou o bom humor (ou não) com uma boa troca de ideias; a Hilton Machado, pela parceria e por quebrar a monotonia dos dias no laboratório; e a Gisliany Alves, pelas boas conversas, pelas risadas e pela amizade que atravessa a cidade para um treino de Muay Thai. Um agradecimento especial a Breno Santos, pelos conselhos e pelo auxílio com os materiais que ajudaram a construir a base teórica deste trabalho.

Aos amigos que sempre se fizeram presentes, não importa a distância.

A todos que, direta ou indiretamente, contribuíram para a minha formação.

Resumo

A crescente presença de sistemas digitais em veículos tem ampliado o número de funcionalidades disponíveis ao usuário. No entanto, dúvidas recorrentes relacionadas ao uso desses recursos, à interpretação de alertas e à execução de procedimentos operacionais básicos ainda exigem a consulta aos manuais do proprietário para que o manuseio do veículo ocorra de forma adequada. Embora a digitalização desses documentos represente um avanço em relação aos suportes físicos, o acesso às informações continua limitado, sobretudo em situações que exigem respostas rápidas e linguagem acessível. Diante desse contexto, este trabalho propõe uma abordagem orientada a agentes de linguagem, fundamentada na técnica de *Retrieval-Augmented Generation* (RAG), com o objetivo de facilitar a consulta especializada a conteúdos técnicos de manuais automotivos. A metodologia abrange etapas de segmentação dos textos em fragmentos coerentes, indexação em base vetorial, elaboração de *prompts* adaptados a um Modelo de Linguagem de Grande Escala e avaliação comparativa de seis variantes de RAG (Convencional, com Gradiente Descendente, *Multi-Query*, *Step-Back*, *Self-RAG* e *Self-RAG* com Gradiente Descendente). Com isso, foi realizado um experimento em que cada variante foi avaliada conforme a estratégia de *LLM-as-a-judge*, na qual um LLM atribuiu pontuações em fidelidade ao contexto, relevância para a pergunta, completude das informações e verificação de segurança. O conjunto de dados utilizado incluiu vinte perguntas, sendo dez formuladas a partir do manual do Volkswagen Polo 2025 e dez do Fiat Argo 2023. Além disso, a similaridade semântica entre pares de respostas foi mensurada por meio do BERTScore. Os resultados indicaram que o *Step-Back* alcançou a maior pontuação média final e liderou em completude e segurança, ao passo que o *Self-RAG* obteve o melhor desempenho em fidelidade e apresentou convergência semântica elevada com sua variante de gradiente. Esses achados sugerem que mecanismos de reformulação, decomposição e autoavaliação elevam tanto a qualidade quanto a consistência das respostas, evidenciando o potencial de arquiteturas adaptativas para aprimorar a assistência técnica em sistemas embarcados.

Palavras-chave: *Retrieval-Augmented Generation*, Assistência Automotiva, Engenharia de *Prompts*, Modelos de Linguagem, *Self-RAG*, Gradiente Descendente.

Abstract

The increasing presence of digital systems in vehicles has expanded the number of functionalities available to users. However, recurring doubts related to the use of these features, the interpretation of alerts, and the execution of basic operational procedures still require consulting the owner's manuals to ensure proper vehicle handling. Although the digitization of these documents represents an advancement over physical formats, access to information remains limited, especially in situations that demand quick responses and accessible language. In this context, this work proposes a language agent–oriented approach, grounded in the Retrieval-Augmented Generation (RAG) technique, with the goal of facilitating specialized consultation of technical content in automotive manuals. The methodology encompasses stages such as segmenting texts into coherent fragments, indexing them in a vector database, crafting prompts tailored to a Large Language Model, and conducting a comparative evaluation of six RAG variants (Conventional, with Gradient Descent, Multi-Query, Step-Back, Self-RAG, and Self-RAG with Gradient Descent). Accordingly, an experiment was conducted in which each variant was evaluated using the LLM-as-a-judge strategy, in which an LLM assigned scores for contextual faithfulness, question relevance, informational completeness, and safety verification. The dataset used comprised twenty questions, ten based on the Volkswagen Polo 2025 manual and ten on the Fiat Argo 2023 manual. Additionally, semantic similarity between pairs of responses was measured using BERTScore. The results indicated that Step-Back achieved the highest overall average score and led in completeness and safety, whereas Self-RAG delivered the best performance in faithfulness and exhibited high semantic convergence with its gradient-based variant. These findings suggest that mechanisms of reformulation, decomposition, and self-assessment improve both the quality and consistency of responses, highlighting the potential of adaptive architectures to enhance technical assistance in embedded systems.

Keywords: Retrieval-Augmented Generation, Automotive Assistance, Prompt Engineering, Language Models, Self-RAG, Gradient Descent.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Abreviaturas

AAOS	*American Academy of Orthopaedic Surgeons*
BERT	*Bidirectional Encoder Representations from Transformers*
BF	*Big Five*
CoT	*Chain-of-Thought*
ELMo	*Embeddings from Language Models*
GloVe	*Global Vectors for Word Representation*
GPT	*Generative Pre-trained Transformer*
IA	Inteligência Artificial
IsREL	Indicador de Relevância dos Fragmentos (*Relevance Indicator Token*)
IsSUP	Indicador de Suporte Factual (*Support Indicator Token*)
IsUSE	Indicador de Utilidade da Resposta (*Usefulness Indicator Token*)
LLM	*Large Language Model*
LSTM	*Long Short-Term Memory*
MBTI	*Myers-Briggs Type Indicator*
METEOR	*Metric for Evaluation of Translation with Explicit ORdering*
NLG	*Natural Language Generation*
NLU	*Natural Language Understanding*
PLN	Processamento de Linguagem Natural
RAG	*Retrieval-Augmented Generation*
RLHF	*Reinforcement Learning from Human Feedback*
RNN	Rede Neural Recorrente
ROUGE	*Recall-Oriented Understudy for Gisting Evaluation*

SBERT *Sentence Bidirectional Encoder Representations from Transformers*

TF–IDF *Term Frequency–Inverse Document Frequency*

Seção 1

Introdução

A Quarta Revolução Industrial, também conhecida como Indústria 4.0, marcou o início de uma era de automação, interconectividade e computação em nuvem, transformando os processos industriais (Kiciński et al. 2021). Como consequência, aumentaram o volume e a complexidade da documentação técnica relacionada ao funcionamento, uso e suporte de sistemas sofisticados, o que elevou os desafios de acesso à informação (Furth 2018). No setor automotivo, um dos pilares dessa transformação, a digitalização de informações técnicas, como manuais de veículos, tem ganhado espaço (Lopez-Vega & Moodysson 2023).

Entretanto, embora a conversão de manuais impressos para formatos digitais represente um avanço, ainda persistem obstáculos de acessibilidade e compreensão por parte dos usuários (Winkelhake 2019). Nesse cenário, métodos baseados em Inteligência Artificial Generativa, como o *Retrieval-Augmented Generation* (RAG), vêm sendo explorados por oferecerem mecanismos de busca orientados ao conteúdo em grandes volumes de documentos técnicos, o que pode favorecer um acesso estruturado à informação (Medeiros et al. 2023).

A dificuldade de acesso a informações técnicas veiculares não se restringe a contextos especializados, alcançando também o cotidiano de condutores em situações rotineiras de uso. Essa limitação se torna maior diante da ausência de mecanismos que permitam localizar com agilidade conteúdos específicos, o que pode comprometer a experiência do usuário (Forster et al. 2019). Em muitos casos, dúvidas relacionadas ao uso de funcionalidades eletrônicas, à interpretação de alertas no painel e à execução de procedimentos básicos, como a verificação do nível de óleo ou a configuração do sistema multimídia, exigem consulta direta aos manuais do proprietário (Atombo et al. 2025). Quando esse acesso não se dá de forma clara, os usuários podem recorrer a fontes informais, adotar soluções inadequadas ou deixar de executar ações necessárias. Como resultado, aspectos como a segurança do veículo, sua vida útil e os custos de manutenção podem sofrer impactos negativos (Saleh et al. 2019, Mena et al. 2024).

Diante dessas limitações, têm sido propostas abordagens baseadas em técnicas de Processamento de Linguagem Natural (PLN), com foco nos Modelos de Linguagem de Grande Escala (em inglês, *Large Language Models*, LLMs). Esses modelos são treinados com grandes volumes de dados textuais, o que lhes permite interpretar consultas realizadas em linguagem natural, mesmo sem correspondências exatas entre os termos pesquisados e os conteúdos armazenados (Thapa et al. 2025). Dessa forma, torna-se pos-

sível oferecer respostas mais próximas das necessidades reais dos usuários, ainda que suas dúvidas sejam formuladas de maneira informal ou pouco estruturada. A partir dessa integração, abre-se a possibilidade de mitigar algumas das dificuldades associadas aos métodos tradicionais de busca textual, especialmente em situações que envolvem o acesso a conteúdos técnicos presentes em manuais automotivos (Medeiros et al. 2023).

Nesse contexto, o RAG busca contornar as limitações dos modelos gerativos ao incorporar, no processo de resposta, informações recuperadas de fontes externas (Han et al. 2024). A técnica combina dois estágios: primeiro, identifica trechos relevantes em uma base documental por meio de busca por similaridade; em seguida, utiliza esse material como suporte para a construção da resposta textual. Ao recorrer a evidências externas, o modelo consegue responder a perguntas com base em conteúdos não disponíveis durante o treinamento (Han et al. 2024). A partir dessa lógica, surgiram variações que ajustam a forma como a recuperação e a geração são coordenadas, incorporando mecanismos capazes de tomar decisões autônomas sobre a reformulação de perguntas, a seleção de evidências ou a organização do fluxo de resposta (Asai et al. 2023). Tais adaptações procuram adequar a abordagem a domínios específicos sem comprometer sua estrutura conceitual (Medeiros et al. 2023).

Entre as estratégias voltadas a esse alinhamento, destaca-se a Engenharia de *Prompt*. Essa técnica envolve a formulação das instruções fornecidas aos modelos de linguagem, com o intuito de alinhar as respostas às necessidades dos usuários e ao contexto de um domínio específico. Com base nessa formulação, é possível favorecer o desempenho de sistemas conversacionais automatizados, conhecidos como *chatbots*. Como consequência, tornam-se viáveis interações mais naturais, compreensíveis e adequadas em ambientes de consulta técnica (Medeiros et al. 2023).

1.1 Problemática e Hipóteses

A consulta aos manuais de veículos é uma forma comum de buscar instruções em situações que envolvem dúvidas sobre operação ou manutenção (Atombo et al. 2025). No entanto, apesar da disponibilidade em formato digital, esses documentos costumam ser extensos e organizados de modo que dificulta a localização rápida de informações específicas (Winkelhake 2019, Forster et al. 2019). Nesses casos, quando o usuário não encontra facilmente o que procura, é possível que adie decisões ou realize procedimentos sem a orientação adequada. Como consequência, aumentam os riscos de falhas e de comprometimento do funcionamento correto do veículo (Saleh et al. 2019, Mena et al. 2024).

Diante desse cenário, foram elaboradas as seguintes questões de pesquisa:

QP 1 - Variações de RAG que incorporam etapas adicionais de processamento produzem respostas com maior alinhamento semântico ao conteúdo de referência?

QP 2 - Como a aplicação de Engenharia de *Prompts* influencia a qualidade das respostas geradas em consultas baseadas em manuais técnicos?

QP 3 - Até que ponto a integração dessas técnicas pode superar as limitações de abordagens convencionais no setor automotivo?

A partir dessas questões, estabelecem-se as seguintes hipóteses:

Hipótese nula ($\mathbf{H}_0$): variações na arquitetura do RAG e nas estratégias de Engenharia de *Prompts* não produzem melhorias significativas na qualidade das respostas geradas por um *chatbot* aplicado a manuais técnicos.

Hipótese alternativa ($\mathbf{H}_1$): variações na arquitetura do RAG e nas estratégias de Engenharia de *Prompts* produzem melhorias significativas na qualidade das respostas geradas por um *chatbot* aplicado a manuais técnicos.

1.2 Objetivos da Pesquisa

Para a realização desta pesquisa, definem-se o objetivo geral e os objetivos específicos, conforme apresentados a seguir.

1.2.1 Objetivo Geral

O objetivo geral deste estudo é investigar como variações da técnica de RAG, combinadas a estratégias de Engenharia de *Prompts*, influenciam a qualidade das respostas geradas por um *chatbot* aplicado a manuais técnicos automotivos. O foco está em analisar o desempenho dessas variações por meio de métricas quantitativas e avaliações semânticas, tendo como referência a qualidade das respostas em relação ao conteúdo técnico presente nos manuais automotivos.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Implementar e testar diferentes variantes da técnica de RAG, aplicadas a manuais técnicos em formato PDF;
- Avaliar o desempenho das variantes testadas, considerando os efeitos das modificações estruturais e das estratégias de Engenharia de *Prompts* na qualidade das respostas;
- Analisar os resultados com base em métricas e avaliações semânticas, considerando a aderência ao conteúdo técnico de referência.

1.3 Justificativa

A digitalização de manuais veiculares tem evoluído como parte do processo de transformação digital, mas esse movimento não tem sido acompanhado por mudanças estruturais na forma como os documentos são organizados para consulta (Winkelhake 2019, Lopez-Vega & Moodysson 2023). Em muitos casos, o conteúdo permanece disposto de

maneira linear, com poucas opções de navegação ou mecanismos que favoreçam a recuperação pontual da informação (Furth 2018). Esse cenário pode comprometer o uso prático do material, sobretudo quando há necessidade de localizar instruções específicas em contextos que exigem agilidade ou envolvem termos técnicos pouco familiares (Forster et al. 2019).

Em decorrência disso, a dificuldade de acesso se agrava quando a busca por informações depende da formulação exata de palavras ou expressões (Winkelhake 2019). A ausência de recursos que interpretem a intenção por trás da consulta limita a recuperação textual, mesmo quando a documentação contém os dados necessários. Nessa condição, a estrutura tradicional de busca torna-se pouco sensível às variações linguísticas e às formas naturais de questionamento, o que reduz a autonomia de quem precisa utilizar o material sem formação técnica prévia (Forster et al. 2019).

Nesse contexto, torna-se oportuno investigar abordagens que articulem modelos de linguagem com mecanismos de recuperação por similaridade. A técnica de RAG propõe justamente essa integração, permitindo que respostas sejam elaboradas com base em trechos recuperados de fontes externas (Han et al. 2024). Com isso, o sistema passa a construir respostas que dialogam com o conteúdo documental sem exigir correspondência literal entre consulta e referência. Ao explorar variações dessa técnica, assim como estratégias de Engenharia de *Prompts*, esta pesquisa busca compreender os efeitos dessas escolhas sobre a qualidade das respostas geradas, sobretudo em domínios com documentação técnica extensa (Asai et al. 2023).

Dessa forma, a proposta se justifica pela necessidade de aprimorar a mediação entre linguagem natural e conteúdo técnico, em especial nos casos em que o acesso à informação condiciona decisões relacionadas à operação, ao diagnóstico de falhas ou à realização de procedimentos. Ao concentrar-se na análise de técnicas capazes de reorganizar o processo de consulta sem alterar a estrutura dos documentos, a pesquisa pode contribuir para o desenvolvimento de soluções mais adaptadas às exigências do uso prático da informação técnica.

1.4 Estrutura do Projeto

Esta seção apresentou a motivação principal para o desenvolvimento deste trabalho, incluindo a contextualização da pesquisa, a definição da problemática, as hipóteses, os objetivos e a justificativa, proporcionando, assim, uma visão ampla sobre o contexto de aplicação. Em sequência, a Seção 2 abordará os conceitos e termos essenciais para o entendimento do estudo. Além disso, a Seção 3 discutirá pesquisas relacionadas, destacando, por sua vez, as diferenças em relação ao estudo proposto. Na sequência, a Seção 4 detalhará os aspectos metodológicos do trabalho, incluindo a descrição das abordagens baseadas em RAG que estão sendo propostas. Posteriormente, a Seção 5 descreverá o experimento realizado para avaliar as abordagens propostas. Em seguida, a Seção 6 apresentará os resultados obtidos com a aplicação da metodologia. Finalmente, a Seção 7 discutirá as conclusões do estudo, as limitações observadas, as contribuições alcançadas e possíveis desdobramentos futuros.

Seção 2

Fundamentação Teórica

Nesta seção, apresentam-se os principais conceitos que sustentam este estudo. Inicialmente, são discutidos os *chatbots*, com ênfase em sua evolução como sistemas de diálogo e na forma como se inserem no campo da Inteligência Artificial. Em seguida, aborda-se o Processamento de Linguagem Natural, destacando-se os Modelos de Linguagem de Grande Escala, cuja alta capacidade de parametrização amplia o potencial de geração textual. Na sequência, descreve-se a arquitetura *Transformer*, com foco nos mecanismos de atenção, e introduz-se a família dos *Generative Pre-trained Transformers* (GPT).

O texto prossegue com a Engenharia de *Prompt*, área voltada à formulação de instruções que guiam a atuação dos modelos. Nesse contexto, o *chunking* é apresentado como uma técnica destinada à divisão de documentos em segmentos menores, enquanto a Busca por Similaridade é abordada como estratégia para recuperação de conteúdos relacionados. Em seguida, introduzem-se os *embeddings*, utilizados para representar informações com base em relações semânticas. Além disso, exploram-se as Bases de Dados Vetoriais, com destaque para o sistema Milvus, e discute-se o uso do Gradiente Descendente como método de ajuste e refinamento. Por fim, descreve-se a técnica de *Retrieval-Augmented Generation* e sua versão com agentes, conhecida como *Agentic* RAG.

2.1 *Chatbots*

A busca por formas mais eficientes de interação entre humanos e sistemas computacionais tem impulsionado, ao longo das últimas décadas, o desenvolvimento de interfaces automatizadas baseadas em linguagem natural (Jurafsky & Martin 2023). Nesse cenário, os *chatbots* destacam-se como agentes conversacionais projetados para simular o diálogo humano, funcionando como intermediários entre usuários e serviços digitais. Esses sistemas vêm sendo utilizados de forma abrangente em contextos diversos, pois proporcionam respostas rápidas e garantem disponibilidade contínua (Gupta et al. 2020). Com os avanços recentes da Inteligência Artificial e, em especial, do Processamento de Linguagem Natural, os *chatbots* passaram de modelos baseados em regras fixas para sistemas mais dinâmicos e adaptativos. Como resultado, tornaram-se capazes de promover interações mais naturais, contextualizadas e eficazes, aproximando as respostas geradas pela máquina da comunicação humana (Adamopoulou & Moussiades 2020).

O conceito de *chatbot* remonta à década de 1960, com o surgimento do programa

ELIZA, desenvolvido por Joseph Weizenbaum. Embora limitado, esse sistema pioneiro já buscava simular interações verbais com usuários em um contexto psicoterapêutico (Adamopoulou & Moussiades 2020, Jurafsky & Martin 2023). Desde então, os avanços em Inteligência Artificial e, especialmente, em Processamento de Linguagem Natural impulsionaram o surgimento de sistemas mais sofisticados, capazes de compreender intenções e produzir respostas contextualmente adequadas (Adamopoulou & Moussiades 2020). Nesse processo evolutivo, os *chatbots* passaram a ser classificados em duas categorias principais: os baseados em regras, que operam por meio de fluxos condicionais e respostas pré-programadas, e os baseados em IA, que utilizam algoritmos de aprendizado de máquina para gerar respostas de forma mais autônoma (Julianto et al. 2024). Esta última abordagem emprega de forma abrangente modelos de linguagem da família *Transformer*, como o *Generative Pre-trained Transformer*, os quais proporcionam maior flexibilidade, personalização e coerência nas interações (Julianto et al. 2024).

Entre as diferentes formas de aplicação desses sistemas, destacam-se os *chatbots* voltados à recuperação de informações, nos quais o foco está na identificação da intenção expressa pelo usuário e na produção de respostas pontuais com base em dados previamente estruturados ou documentos externos. Nesses casos, mais do que simular uma conversa contínua, o objetivo está em oferecer informações relevantes de maneira clara, objetiva e tecnicamente precisa. Esse tipo de aplicação é comum em cenários que exigem precisão na resposta e aderência ao conteúdo original, como ocorre em sistemas de suporte técnico, consultas especializadas e serviços baseados em documentação (Julianto et al. 2024).

2.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural é um campo da Inteligência Artificial relacionado ao desenvolvimento de métodos que tornam as máquinas capazes de compreender e produzir linguagem humana (Cai et al. 2025). Sua importância está relacionada ao fato de que a linguagem é um dos principais meios pelos quais as pessoas se expressam, formulam ideias e interagem com o mundo ao seu redor. Por isso, compreender como representar, interpretar e gerar linguagem natural de maneira clara e contextualizada é importante para construir sistemas que realmente contribuam para que haja uma boa comunicação entre usuários e tecnologias digitais (Lane & Dyshel 2025).

Historicamente, o PLN teve origem na interseção entre a linguística e a ciência da computação, com a finalidade inicial de automatizar a aplicação de regras gramaticais para a interpretação automática de textos (Jurafsky & Martin 2023). Com o avanço da capacidade computacional, essa área evoluiu e passou a adotar métodos estatísticos que permitiram às máquinas identificar padrões em grandes volumes de textos. Posteriormente, o advento de técnicas baseadas em aprendizado profundo proporcionou avanços adicionais ao possibilitar uma melhor representação dos contextos e significados das palavras (Lane & Dyshel 2025).

Nesse contexto, destacam-se duas áreas fundamentais dentro do PLN: a compreensão (*Natural Language Understanding* – NLU) e a geração de linguagem natural (*Natural Language Generation* – NLG) (Lane & Dyshel 2025). A compreensão refere-se à capa-

cidade dos sistemas computacionais interpretarem e representarem o significado contido em textos. Essa tarefa envolve técnicas para classificação de intenções, reconhecimento de entidades nomeadas, análise de sentimentos e recuperação semântica de informações. A base técnica dessa capacidade reside na transformação do texto em representações numéricas (vetores), possibilitando o uso de algoritmos de aprendizado de máquina para inferir significados e contextos (Lane & Dyshel 2025), como está representado na Figura 2.1.

Figura 2.1: Codificação da linguagem natural.

Fonte: Adaptada de Lane & Dyshel (2025).

Por outro lado, a geração de linguagem natural envolve técnicas destinadas à produção automática de textos coerentes a partir de informações estruturadas ou representações internas em formato vetorial. Por meio do processo de decodificação, conforme apresentado na Figura 2.2, esses modelos são capazes de transformar dados numéricos em textos inteligíveis, utilizados por sistemas que geram respostas automáticas, relatórios sintéticos ou assistentes virtuais. Em particular, essa capacidade é determinante para a qualidade das respostas produzidas por sistemas conversacionais, como os *chatbots* (Lane & Dyshel 2025).

Figura 2.2: Decodificação na geração de linguagem natural.

Fonte: Adaptada de Lane & Dyshel (2025).

Ao considerar essas duas vertentes de forma integrada, percebe-se que a qualidade da comunicação mediada por máquinas depende da capacidade de interpretar adequadamente as mensagens humanas e de formular respostas coerentes, contextualizadas e compreensíveis para o interlocutor. Desse modo, o PLN se aproxima da linguagem cotidiana e contribui para tornar as interações com sistemas computacionais mais naturais e acessíveis (Lane & Dyshel 2025).

2.2.1 Modelos de Linguagem de Grande Escala

Modelos de Linguagem de Grande Escala transformaram a forma como se processa a linguagem natural (Wu et al. 2024). Em contraste com abordagens anteriores, mais limitadas a tarefas como classificação de e-mails ou identificação de padrões simples, os LLMs permitiram avanços em tarefas que envolvem linguagem complexa (Jurafsky

& Martin 2023, Wu et al. 2024). Esses modelos lidam com tradução, resumo, geração de textos e respostas a perguntas de maneira contextualizada, oferecendo saídas mais próximas da comunicação humana e superando restrições que dificultavam o uso mais amplo de sistemas computacionais em linguagem natural (Raschka 2025).

A expressão "grande escala" abrange, portanto, dois aspectos principais: a dimensão dos dados empregados no treinamento — que frequentemente envolvem bilhões de palavras extraídas de fontes diversas — e a complexidade do modelo, que pode operar com centenas de bilhões de parâmetros (Raschka 2025). Esse volume de informação e poder computacional permite identificar padrões linguísticos sutis e estabelecer relações contextuais de longo alcance. Com isso, torna-se possível superar limitações observadas em gerações anteriores de modelos de linguagem. Esse avanço aparece com mais nitidez em tarefas que exigem compreensão semântica mais elaborada ou produção de texto com coerência discursiva (Raschka 2025).

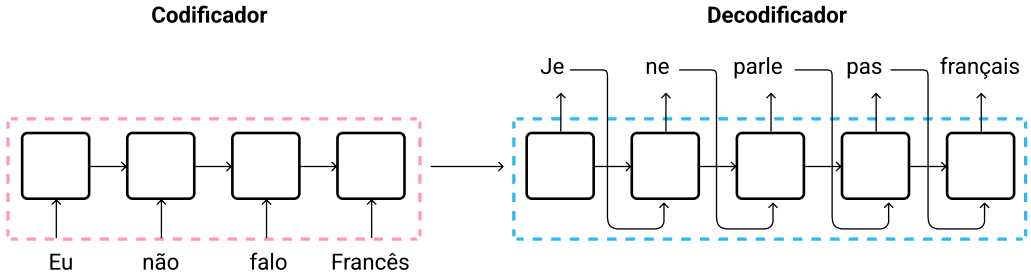
Grande parte desse progresso está associada à adoção de uma arquitetura específica, conhecida como *Transformer*. Diferentemente dos modelos sequenciais tradicionais, os *Transformers* empregam mecanismos de atenção que atribuem pesos variáveis às partes da entrada textual durante o processamento, gerando representações contextuais mais refinadas (Vaswani et al. 2017). Essa abordagem favoreceu o desenvolvimento de modelos versáteis, capazes de gerar texto com fluidez e coerência, mesmo em situações em que a resposta demanda inferência, manutenção de contexto ou adaptação ao estilo da entrada (Raschka 2025).

2.2.2 Arquitetura *Transformer*

O surgimento da arquitetura *Transformer* representou uma mudança decisiva no campo do Processamento de Linguagem Natural, ao oferecer uma alternativa mais eficiente e paralelizável em relação aos modelos recorrentes (Vaswani et al. 2017, Raschka 2025). Antes disso, as Redes Neurais Recorrentes (RNNs) e suas variantes, como as redes *Long Short-Term Memory* (LSTM), eram empregadas em tarefas sequenciais, como tradução automática e análise de sentimentos (Vaswani et al. 2017, Raschka 2025). Esses modelos processavam os dados palavra por palavra, preservando um estado interno que se atualizava continuamente ao longo da sequência. Embora funcionassem de forma adequada em contextos mais curtos, essa estrutura sequencial impunha a necessidade de processar os elementos na ordem em que apareciam, como ilustrado na Figura 2.3. Essa característica dificultava o aproveitamento de relações de longo alcance e trazia instabilidades durante o treinamento, especialmente devido à ocorrência de gradientes que desapareciam ou se tornavam excessivamente grandes (Raschka 2025).

Com a introdução do *Transformer*, essas restrições foram superadas por meio de uma nova estrutura que processa todos os elementos da sequência simultaneamente. Conforme ilustrado na Figura 2.4, a arquitetura adota um modelo codificador-decodificador composto por múltiplas camadas idênticas, cada uma formada por mecanismos de atenção, redes neurais *feedforward*, normalização e conexões residuais (Vaswani et al. 2017). Embora essa seja a configuração original, diferentes modelos derivados do *Transformer* utilizam variações dessa estrutura. Por exemplo, arquiteturas como o *Bidirectional Enco-*

Figura 2.3: Visão de alto nível do fluxo sequencial em uma Rede Neural Recorrente.



Fonte: Adaptada de Raschka (2025).

der Representations from Transformers (BERT) empregam exclusivamente o codificador, enquanto modelos da família GPT utilizam apenas o decodificador. Essas adaptações decorrem da natureza das tarefas que cada modelo busca resolver (Raschka 2025).

O codificador é o primeiro bloco responsável por processar a sequência de entrada. Cada palavra é representada por um vetor numérico, conhecido como *embedding*, que captura informações semânticas e sintáticas básicas. No entanto, como o *Transformer* não possui uma estrutura sequencial, ele não tem acesso à ordem dos *tokens* de forma natural. Para resolver isso, adiciona-se aos *embeddings* uma codificação posicional (*positional encoding*), que insere informações sobre a posição de cada elemento na sequência. Essa codificação pode ser obtida por meio de funções seno e cosseno com diferentes frequências (Raschka 2025).

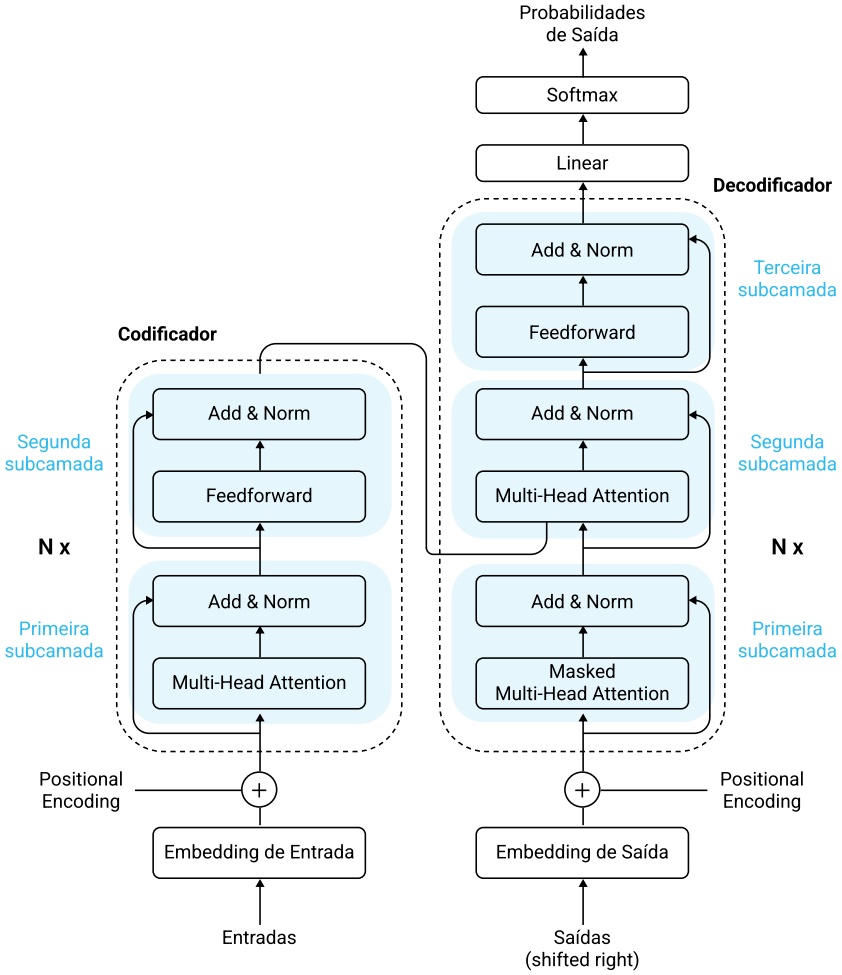
A seguir, esses vetores passam por um componente da arquitetura denominado mecanismo de atenção. Esse mecanismo permite que o modelo analise, para cada palavra, quais outras palavras da sequência são mais relevantes para seu entendimento. Para isso, ele utiliza três vetores derivados da entrada: consulta (*query*), chave (*key*) e valor (*value*). O vetor de consulta de uma palavra é comparado com os vetores de chave de todas as outras palavras, produzindo um conjunto de pesos que indicam a importância relativa de cada uma. Esses pesos são, então, aplicados aos vetores de valor, gerando uma nova representação para a palavra em questão, ajustada ao seu contexto (Raschka 2025).

A operação matemática principal que define esse processo é conhecida como atenção escalada por produto escalar, expressa conforme a Equação 2.1. Nessa fórmula, Q, K e V são as matrizes que contêm os vetores de consulta, chave e valor, respectivamente, e d_k representa a dimensionalidade dos vetores de chave. O fator de normalização $\sqrt{d_k}$ evita que os valores se tornem excessivamente grandes, garantindo a estabilidade da função *softmax* (Raschka 2025).

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

Para ampliar a capacidade de capturar diferentes tipos de relações contextuais, o *Transformer* aplica esse cálculo em paralelo em várias projeções lineares distintas, um processo conhecido como atenção com múltiplas cabeças (*multi-head attention*). Cada "cabeça" é responsável por identificar padrões diferentes, o que enriquece a representação final. Os resultados dessas múltiplas atenções são concatenados e passam por uma

Figura 2.4: Arquitetura *Transformer*.



Fonte: Adaptada de Raschka (2025).

projeção final, compondo a saída da camada de atenção (Raschka 2025).

Após a autoatenção (*self-attention*), os vetores seguem para uma rede neural do tipo *feedforward*, aplicada individualmente a cada posição da sequência. Essa rede, composta por duas camadas lineares intercaladas por uma função de ativação, é responsável por transformar os dados em um espaço de maior complexidade. Entre cada um desses blocos, atenção e *feedforward*, são adicionadas conexões residuais e normalização de camada, o que facilita o aprendizado e estabiliza o treinamento (Raschka 2025).

O decodificador apresenta uma estrutura semelhante à do codificador, com duas modificações que adaptam seu funcionamento à tarefa de geração textual. A primeira está na autoatenção mascarada (*masked multi-head attention*), que impede cada posição da saída de acessar palavras futuras. Essa restrição assegura a produção ordenada do texto e mantém a linearidade da linguagem. A segunda modificação introduz um bloco de atenção cruzada (*multi-head attention*), que permite ao modelo utilizar as representações extraídas pelo codificador. Dessa forma, o sistema estabelece correspondências entre a entrada e a saída e gera respostas alinhadas ao conteúdo original (Raschka 2025).

Dessa forma, a composição do *Transformer* tornou-se adequada para diversas tarefas no PLN, pois sua arquitetura favorece o paralelismo durante o treinamento, proporciona controle sobre o fluxo de informação entre as camadas e permite adaptação a diferentes tipos de entrada. Em razão dessas propriedades, o *Transformer* passou a servir de base para modelos com maior capacidade de representação, como os Modelos de Linguagem de Grande Escala (Raschka 2025).

2.2.3 *Generative Pre-trained Transformer*

O *Generative Pre-trained Transformer* surgiu nos laboratórios da OpenAI e ganhou projeção pública após o lançamento do ChatGPT (OpenAI 2023). O modelo adota apenas o bloco decodificador do *Transformer* e aprende, na etapa de pré-treinamento, a prever o próximo *token* de extensos volumes de texto. Essa tarefa, chamada de modelagem autoregressiva, faz com que a probabilidade de uma sequência seja decomposta como o produto das probabilidades condicionais de cada palavra. Cada termo é calculado com base no histórico anterior, o que disciplina o modelo a prosseguir palavra a palavra, seguindo uma ordem causal (Radford et al. 2018, Brown et al. 2020).

A sigla GPT passou a denominar uma família de versões numeradas (GPT-1, GPT-2, GPT-3, GPT-3.5 e GPT-4), todas baseadas no decodificador, embora diferenciadas pela escala de parâmetros, pela quantidade e diversidade do corpus (conjunto de texto) usado no pré-treinamento e pelos métodos de alinhamento empregados após essa fase (Brown et al. 2020, OpenAI 2023). A incorporação de *Instruction Tuning*, *Reinforcement Learning from Human Feedback* (RLHF) e, no caso do GPT-4, a capacidade multimodal para entrada de imagens reflete a ampliação progressiva de escopo e desempenho desses modelos (Ouyang et al. 2022, OpenAI 2023).

Entre as limitações mais discutidas, está a forma como o mecanismo de atenção lida com sequências extensas (Zhang et al. 2024). À medida que o número de palavras na entrada aumenta, o esforço computacional cresce de forma desproporcional, já que o modelo precisa considerar todas as combinações possíveis entre os elementos da sequência. Isso compromete a viabilidade do uso em tarefas que exigem janelas de contexto muito longas (Zhang et al. 2024). Além disso, o conhecimento do GPT permanece limitado ao período coberto pelos dados com os quais foi treinado, o que impede que ele incorpore eventos mais recentes sem intervenções adicionais (OpenAI 2023). Há também o risco de o modelo produzir respostas imprecisas ou enviesadas, reflexo tanto das características do conjunto de dados original quanto do modo como ele seleciona palavras ao gerar um texto (OpenAI 2023).

Apesar de suas restrições, o GPT apresenta desempenho consistente em diferentes tarefas de linguagem natural. Com poucas instruções em linguagem cotidiana, o modelo consegue gerar textos articulados, responder perguntas, traduzir passagens, produzir resumos e escrever trechos de código (Brown et al. 2020). Esses resultados motivaram estudos voltados ao aprimoramento de sua utilização, incluindo estratégias para alinhamento com preferências humanas, compactação de parâmetros, identificação de erros factuais e exploração de usos em contextos como educação, apoio à escrita e pesquisa (Bommasani et al. 2021, Ouyang et al. 2022). Nesse cenário, o GPT passou a integrar um conjunto

de modelos com ampla presença em aplicações que dependem de compreensão e geração textual (Bommasani et al. 2021).

2.2.4 Engenharia de *Prompt*

Com a ampla adoção de LLMs, surgiu a necessidade de compreender de que modo suas respostas podem ser orientadas sem recorrer a mudanças em sua estrutura treinada (Wu et al. 2024). Nesse contexto, a Engenharia de *Prompt* corresponde à prática de elaborar a entrada textual fornecida ao modelo. Seu objetivo é guiar a saída por meio das instruções presentes na própria solicitação. Diante dessas condições, em que o modelo já se encontra treinado e não pode ser ajustado de forma direta, a elaboração da entrada passa a desempenhar um papel importante. Essa etapa contribui para adaptar o sistema às demandas de uso, contornando restrições técnicas ou limitações de acesso ao processo de desenvolvimento (Huyen 2025).

Um *prompt* pode ser composto por diferentes partes, como a descrição da tarefa, a atribuição de um papel ao modelo, a definição do formato da resposta e, em alguns casos, exemplos que indicam o tipo de saída desejada (Wu et al. 2024, Huyen 2025). Quando esses exemplos fazem parte da própria entrada textual, a prática é conhecida como *in-context learning*, processo no qual o modelo identifica padrões a partir de instâncias presentes na solicitação. Na ausência de exemplos, utiliza-se a expressão *zero-shot*; quando a entrada inclui uma ou mais ilustrações da tarefa, adota-se a designação *few-shot* (Huyen 2025).

A depender da composição adotada, a forma como o *prompt* é estruturado pode influenciar o modo como o modelo interpreta e executa a tarefa (Lu et al. 2022, Huyen 2025). Instruções posicionadas no início do texto tendem a favorecer a leitura do comando, enquanto alterações discretas na redação ou na ordem dos blocos informativos podem levar a saídas distintas. Nesse cenário, aspectos como clareza na formulação, sequência das instruções e coerência entre enunciado e propósito da tarefa são considerados determinantes para o desempenho observado (Huyen 2025).

Para além da estrutura básica, certas tarefas exigem abordagens mais refinadas na elaboração do *prompt*. Quando há múltiplas etapas envolvidas, é comum a adoção de estratégias baseadas na fragmentação do processo em blocos menores, cada qual acompanhado de instruções próprias. Essa divisão permite delimitar com maior precisão as ações esperadas, facilitar ajustes pontuais e reorganizar as etapas conforme as necessidades de uso, sem comprometer a integridade do conjunto (Huyen 2025).

Em contextos nos quais se busca explicitação do raciocínio, uma técnica frequentemente aplicada é a chamada *Chain-of-Thought*, que orienta a resposta por meio de uma sequência ordenada de justificativas ou operações (Wei et al. 2023). Esse tipo de instrução tende a organizar a saída de forma mais compreensível, além de reduzir inconsistências ao longo do processo. Há também cenários em que o modelo é instruído a utilizar apenas as informações fornecidas na entrada, o que pode evitar interferências indesejadas do conteúdo assimilado durante o pré-treinamento. Tais práticas são recorrentes em aplicações que operam com dados restritos, fontes controladas ou exigem aderência a simulações específicas (Huyen 2025).

Por fim, a durabilidade e a consistência dos *prompts* ao longo do tempo dependem

do acompanhamento das modificações introduzidas nas versões dos modelos utilizados. Alterações na arquitetura ou no comportamento padrão podem impactar a interpretação de instruções anteriormente estáveis. Em aplicações abertas, é preciso considerar ainda os riscos associados à exposição de conteúdos sensíveis, à manipulação por instruções maliciosas e à quebra de restrições previstas no projeto inicial (Huyen 2025).

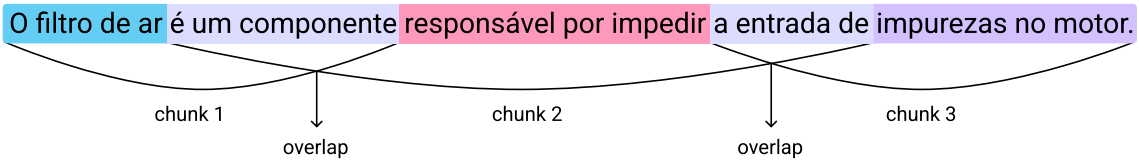
2.2.5 *Chunking*

Em sistemas baseados em linguagem natural, a manipulação de documentos extensos exige procedimentos que permitam organizar o conteúdo em unidades mais manejáveis (Lewis et al. 2020). Um desses procedimentos é conhecido como *chunking*, termo que designa a divisão de um texto em blocos menores, chamados *chunks* (Gao et al. 2023). Cada *chunk* corresponde a um segmento contínuo de texto que pode assumir diferentes tamanhos, conforme os critérios adotados na segmentação. Essa divisão busca preservar a coesão interna de cada trecho, ao mesmo tempo em que adapta o material às exigências computacionais do sistema que o processará (Huyen 2025).

A necessidade de aplicar *chunking* decorre, entre outros fatores, das limitações técnicas impostas pelos modelos de linguagem. Tais modelos operam com uma janela de contexto que define a quantidade máxima de *tokens* que podem ser considerados em uma única solicitação. Quando o texto ultrapassa esse limite, parte do conteúdo é ignorada, o que compromete o desempenho do sistema. Mesmo que o volume total de *tokens* esteja dentro da faixa permitida, informações posicionadas em trechos intermediários podem não ser devidamente interpretadas. Nesse cenário, a segmentação em blocos menores oferece uma alternativa viável para garantir que os trechos mais informativos sejam priorizados durante o processamento (Wang et al. 2024, Huyen 2025).

Além de contribuir para a adequação do conteúdo aos limites do modelo, o *chunking* favorece o processo de busca e recuperação textual. Ao organizar o material em unidades mais curtas, torna-se possível realizar buscas mais localizadas, reduzindo a dispersão da informação e facilitando a identificação de passagens relevantes. A segmentação, nesse sentido, não apenas prepara o texto para o modelo, como também estrutura a base de dados sobre a qual se aplicam as estratégias de consulta (Wang et al. 2024, Huyen 2025).

Figura 2.5: Exemplo de *chunks* com sobreposição.



Fonte: Elaborada pela autora (2025).

Para atender a essas finalidades, diferentes métodos de segmentação foram desenvolvidos. Um dos mais utilizados é a segmentação por tamanho fixo, que pode ser definida com base na contagem de caracteres ou *tokens* (Huyen 2025, Gao et al. 2023). Quando

se utiliza o número de caracteres, a separação é feita por meio de delimitadores específicos, como quebras de linha. Em geral, também se estabelece uma sobreposição entre os blocos, o que permite preservar parte do contexto entre segmentos sucessivos e minimizar perdas de continuidade, como ilustra a Figura 2.5. Quando a medida adotada é o número de *tokens*, a segmentação passa a se alinhar de forma mais direta à lógica interna do modelo, que processa o texto com base nessas unidades mínimas (Wang et al. 2024, Huyen 2025).

Em algumas situações, a aplicação de um único critério de separação não é suficiente para garantir o tamanho desejado dos blocos. Nesses casos, métodos recursivos podem ser empregados, com o uso de uma lista ordenada de delimitadores aplicados em sequência. Esse tipo de abordagem permite adaptar a segmentação à estrutura do conteúdo, criando blocos que respeitam tanto a extensão máxima quanto os agrupamentos internos do texto (Wang et al. 2024, Huyen 2025).

Por outro lado, quando o documento apresenta uma organização explícita — como códigos-fonte, páginas HTML ou textos com marcação hierárquica —, é possível aplicar estratégias que consideram essa estrutura preexistente. Nesse caso, os blocos são definidos com base em seções naturais do conteúdo, como cabeçalhos ou blocos funcionais. Essa prática, por se ajustar à forma do documento, é muitas vezes chamada de segmentação adaptativa e pode preservar agrupamentos semânticos já estabelecidos (Huyen 2025).

Além dessas abordagens, existem métodos que buscam formar os *chunks* com base no significado do conteúdo (Gao et al. 2023, Huyen 2025). A segmentação semântica parte da identificação de similaridades entre sentenças ou parágrafos e organiza o texto de modo que trechos com sentido próximo permaneçam juntos. Essa técnica depende da aplicação de representações vetoriais, como os *embeddings*, e ainda se encontra em fase de exploração. No entanto, abre espaço para formas de segmentação mais sensíveis ao conteúdo informacional do texto, aproximando-se do modo como um leitor humano poderia agrupar as ideias (Huyen 2025).

Diante da variedade de estratégias disponíveis, a escolha do método de segmentação depende do tipo de material analisado, da tarefa a ser realizada e do perfil das consultas que se pretende atender. Quando a atividade envolve perguntas diretas e localizadas, blocos menores tendem a oferecer melhores resultados. Já em contextos que exigem maior continuidade, como resumos ou análises temáticas, segmentos mais amplos podem preservar melhor o fluxo das ideias (Wang et al. 2024, Huyen 2025).

2.2.6 Busca por Similaridade

A busca por similaridade é um componente importante em aplicações de PLN que envolvem recuperação de informações, recomendação de conteúdos e sistemas de resposta automática (Karpukhin et al. 2020, Lane & Dyshel 2025). Em vez de depender exclusivamente da correspondência literal entre palavras-chave, essa abordagem considera o grau de semelhança entre representações internas dos textos, geralmente vetores numéricos, o que permite identificar conteúdos semanticamente próximos, mesmo quando formulados com vocabulário diferente (Lane & Dyshel 2025).

Tradicionalmente, técnicas como o modelo *bag-of-words*, que representa um texto

com base na frequência de cada palavra, desconsiderando ordem e contexto, foram muito utilizadas. Outra técnica frequente é o *Term Frequency–Inverse Document Frequency* (TF–IDF), que atribui pesos às palavras de acordo com sua recorrência em diferentes documentos (Manning et al. 2009, Lane & Dyshel 2025). No entanto, essas abordagens costumam falhar na tarefa de capturar o significado das palavras, especialmente quando há sinônimos, ambiguidades ou variações de estilo. Por exemplo, duas frases que transmitem a mesma ideia, mas com vocabulário distinto, podem receber representações muito diferentes nesses modelos, o que dificulta a identificação de similaridade entre elas (Lane & Dyshel 2025).

Diante dessa limitação, surgiram representações vetoriais que incorporam aspectos semânticos. Com elas, é possível mapear documentos e sentenças em espaços numéricos contínuos. Nesses espaços, a noção de proximidade semântica se expressa pela distância entre vetores: quanto menor a distância, mais próximos os significados. Entre as principais medidas empregadas nesse tipo de análise, destacam-se a distância euclidiana e a similaridade cosseno (Lane & Dyshel 2025).

A **distância euclidiana** mede a separação geométrica entre dois vetores em um espaço n-dimensional. Sua fórmula é expressa conforme a Equação 2.2. Essa métrica considera tanto a direção quanto a magnitude dos vetores e pode ser aplicada em situações nas quais os dados foram normalizados ou quando se busca capturar o afastamento absoluto entre representações semânticas (Oduntan et al. 2018).

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

Já a **similaridade cosseno** foca na direção dos vetores, avaliando o cosseno do ângulo entre eles. A expressão matemática dessa medida está apresentada na Equação 2.3 (Oduntan et al. 2018).

$$\text{CosSim}(\mathbf{D}_j,\mathbf{Q}) = \frac{\sum_{i=1}^{t} d_{ij}\cdot q_i}{\sqrt{\sum_{i=1}^{t} d_{ij}^2}\cdot\sqrt{\sum_{i=1}^{t} q_i^2}} \tag{2.3}$$

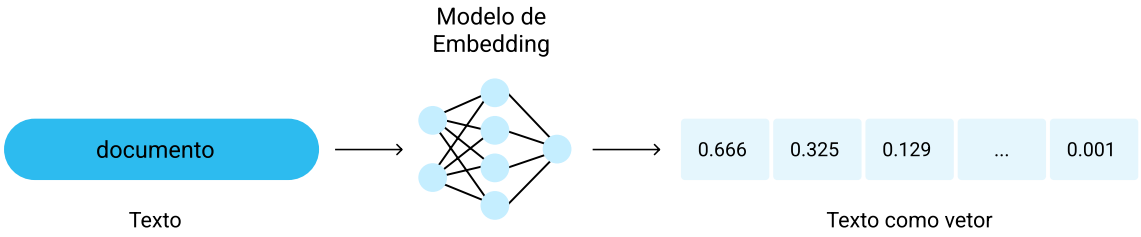
Como não leva em conta a magnitude, a similaridade cosseno costuma ser empregada com vetores esparsos, como os obtidos por meio de TF–IDF (Sitikhu et al. 2019). Em contraste, a distância euclidiana tende a ser usada com representações densas, como os *embeddings* produzidos por modelos de linguagem, já que esses vetores mantêm relações semânticas na forma como se distribuem no espaço vetorial (Karpukhin et al. 2020).

Dessa forma, a busca por similaridade semântica pode ampliar tanto a cobertura quanto a flexibilidade dos sistemas de busca, além de favorecer a recomendação de conteúdos alinhados ao contexto ou interesse do usuário. O bom desempenho desses sistemas decorre da capacidade de representar textos por vetores densos e informativos, que preservam relações semânticas de forma mais robusta (Lane & Dyshel 2025).

2.2.7 *Embeddings*

No PLN, é comum a necessidade de converter textos em estruturas que possam ser manipuladas por algoritmos computacionais (Devlin et al. 2019). Entre as alternativas existentes, destaca-se o uso de representações vetoriais (Figura 2.6), conhecidas como *embeddings*, que transformam palavras, sentenças ou documentos em vetores de dimensão fixa (Mikolov et al. 2013, Boykis 2023). Embora aplicados com frequência a dados textuais, esses vetores também podem ser utilizados para codificar outros tipos de informação, como imagens ou sinais, desde que o objetivo seja organizar conteúdos por proximidade de significado. Cada vetor corresponde a uma sequência de números capaz de capturar traços do conteúdo original, como associações de sentido, contexto de uso e padrões recorrentes (Boykis 2023).

Figura 2.6: Exemplo de *embeddings*.



Fonte: Elaborada pela autora (2025).

Essa representação vetorial oferece uma forma mais densa e informativa em comparação com métodos anteriores, como codificações binárias ou matrizes de contagem. Com os *embeddings*, passa a ser possível posicionar elementos linguísticos em um espaço n-dimensional no qual a proximidade entre vetores reflete, de maneira aproximada, a semelhança de sentido entre os textos representados (Boykis 2023).

Diversas abordagens foram desenvolvidas para a construção desses vetores. Algumas técnicas mais antigas, como Word2Vec, GloVe e FastText, associam vetores fixos às palavras com base em padrões de coocorrência (Peters et al. 2018, Mars 2022). Em contrapartida, métodos mais recentes, como ELMo e BERT, geram representações que variam conforme o contexto, permitindo maior sensibilidade ao uso real da linguagem (Mars 2022).

Embora seja viável treinar modelos próprios, essa etapa demanda conjuntos extensos de dados e recursos computacionais consistentes. Por isso, muitos projetos utilizam modelos pré-treinados, disponibilizados por empresas e comunidades de código aberto. A OpenAI, por exemplo, oferece modelos como *text-embedding-ada-002* e *text-embedding-3-large*, enquanto HuggingFace, Cohere, Voyage, Google e Mistral também mantêm modelos com finalidades diversas e cobertura linguística variada (Mars 2022).

Uma vez gerados, os vetores são geralmente armazenados em estruturas conhecidas como bases vetoriais (*vector databases*), projetadas para facilitar a recuperação de vetores próximos em espaços de alta dimensionalidade. Esses sistemas são otimizados para buscas por similaridade aproximada e podem incluir funcionalidades como ranqueamento

e atualização dinâmica, o que os torna compatíveis com fluxos de consulta em tempo real (Pan et al. 2024).

Dessa forma, os *embeddings* assumem o papel de ponte entre o conteúdo textual original e as operações matemáticas que sustentam modelos de recuperação, classificação e geração. Sua adoção tem permitido o desenvolvimento de soluções capazes de lidar com dados não estruturados de forma compatível com os métodos utilizados em aprendizado de máquina, sem depender de estruturas rígidas de indexação por palavras-chave (Boykis 2023).

2.2.8 Base de Dados Vetorial

Uma vez representados como vetores, os dados textuais precisam ser armazenados de modo que possam ser consultados posteriormente com base em relações de proximidade no espaço vetorial (Gao et al. 2023, Bourne 2024). Essa necessidade leva à incorporação de uma etapa de armazenamento que organiza essas representações numéricas de forma persistente. Após a conversão de documentos em *embeddings*, torna-se necessário guardar essas estruturas em ambientes preparados para buscas sucessivas, nas quais o objetivo é recuperar os trechos mais relacionados ao conteúdo de entrada. Para esse fim, surgiram as bases de dados vetoriais, desenvolvidas para lidar com vetores de alta dimensionalidade (Bourne 2024).

Nesse cenário, esse tipo de base foi concebido para responder a uma necessidade distinta daquela enfrentada por bancos tradicionais (Karpukhin et al. 2020, Bourne 2024). Enquanto estruturas relacionais, como o MySQL ou o PostgreSQL, organizam dados em linhas e colunas, bancos orientados a documentos, como o MongoDB, armazenam objetos semiestruturados. Já as bases vetoriais operam sobre sequências numéricas que codificam conteúdos linguísticos, visuais ou híbridos. O modo como esses vetores são organizados internamente, por meio de estruturas de indexação especializadas, busca reduzir o tempo necessário para identificar, entre milhares ou milhões de vetores, aqueles que apresentam maior proximidade com uma entrada de consulta (Bourne 2024).

Com base nesse princípio, a adoção crescente de *embeddings* em aplicações baseadas em linguagem natural impulsionou o desenvolvimento de diferentes soluções voltadas ao armazenamento vetorial (Gao et al. 2023, Bourne 2024). Algumas dessas foram projetadas desde o início para lidar com esse tipo de dado, como é o caso do Milvus, Qdrant, ChromaDB e Weaviate. Essas ferramentas combinam recursos de banco de dados com mecanismos próprios para busca aproximada e inserção contínua de vetores, além de oferecerem suporte à filtragem por metadados e à separação por coleções (Han et al. 2023, Bourne 2024). Em paralelo, outras abordagens optaram por estender sistemas já consolidados, como o módulo pgvector no PostgreSQL, os recursos de similaridade no Elasticsearch e a integração com vetores no MongoDB Atlas. Essa diversidade de soluções reflete a tentativa de equilibrar desempenho, facilidade de adoção e compatibilidade com arquiteturas existentes (Bourne 2024).

Entre essas alternativas, o Milvus é uma ferramenta que pode ser empregada para o gerenciamento e a consulta de grandes volumes de representações vetoriais (Wang et al. 2021). Trata-se de uma base de dados de código aberto voltada ao armazenamento

e à recuperação de *embeddings*, oferecendo suporte a aplicações de Processamento de Linguagem Natural que dependem da comparação entre vetores. Sua estrutura é compatível com sistemas que exigem consultas rápidas por proximidade, contribuindo para a construção de repositórios que permitem localizar conteúdos relacionados com base em relações de similaridade (Srivamsi et al. 2023).

Por fim, a escolha entre essas alternativas depende do contexto de uso e dos objetivos do sistema (Wang et al. 2021). Ferramentas desenvolvidas exclusivamente para dados vetoriais tendem a apresentar maior controle sobre os algoritmos de indexação, variedade de estratégias de busca e escalabilidade nativa. Já extensões implementadas sobre bancos existentes podem facilitar a integração em ambientes com infraestrutura consolidada, mesmo que imponham restrições quanto à flexibilidade ou ao volume de dados suportado com estabilidade (Bourne 2024).

2.2.9 Gradiente Descendente

Conforme Pryzant et al. (2023), o Gradiente Descendente pode ser aplicado à otimização de *prompts* em LLMs como um processo iterativo baseado em passos semânticos. Esse processo é descrito matematicamente nas Equações (2.4) a (2.8):

1) **Avaliação do *Prompt*** – Dado um *prompt* inicial p_0 e um lote de dados $D = \{x_1, x_2, \ldots, x_n\}$, o modelo M avalia o comportamento de p_0 no lote de entrada e gera uma saída y_i correspondente a cada x_i. A avaliação é definida na Equação 2.4:

$$y_i = M(p_0, x_i), \quad \forall x_i \in D \tag{2.4}$$

2) **Sinal de Perda** (∇) – O gradiente textual g é gerado ao calcular um "sinal de perda" baseado nos erros e_i observados no comportamento do *prompt* p_0. Este erro é definido como mostrado na Equação 2.5:

$$e_i = L(y_i, y_i^{\text{target}}) \tag{2.5}$$

onde L é a função de perda que mede a discrepância entre y_i (saída do modelo) e y_i^{target} (saída esperada ou alvo).

O gradiente semântico g é, então, gerado como um resumo em linguagem natural dos erros, conforme a Equação 2.6:

$$g = \text{Resumo}(p_0, \{e_1, e_2, \ldots, e_n\}) \tag{2.6}$$

3) **Edição do *Prompt*** (δ) – O *prompt* é atualizado ao aplicar δ, que utiliza g para editar p_0 na direção oposta ao gradiente. Esta etapa é descrita na Equação 2.7:

$$p_{t+1} = \text{Edit}(p_t, -g) \tag{2.7}$$

onde Edit é uma função que ajusta o *prompt* p_t corrigindo os problemas indicados por g.

4) **Iteração** – O processo é iterado até que um critério de convergência seja atingido.

O critério pode ser definido como representado na Equação 2.8:

$$\|g\| < \varepsilon \tag{2.8}$$

onde ε é um limiar pré-definido para o gradiente textual.

Essa formulação representa a adaptação do Gradiente Descendente tradicional para um espaço semântico em *prompts*. A ideia central é que, em vez de otimizar parâmetros de um modelo, como ocorre no Aprendizado de Máquina tradicional, o processo otimiza diretamente a formulação textual do *prompt* p_t (Pryzant et al. 2023).

2.2.10 *Retrieval-Augmented Generation*

Retrieval-Augmented Generation é uma técnica que integra mecanismos de recuperação de dados com modelos generativos para melhorar a geração textual (Shahade & Deshmukh 2024). Essa abordagem busca superar a limitação dos modelos de linguagem, os quais dependem de dados de treinamento fixos e, por isso, podem gerar informações desatualizadas ou incompletas. Para que isso seja possível, o sistema recorre a fontes de informação externas (Li, Wang, Wang, Hung, Xie & Wang 2025).

A estratégia de recuperação seguida de geração foi apresentada por Chen et al. (2017) como forma de responder a perguntas em domínios abertos. O processo envolve, em um primeiro momento, a identificação de páginas da Wikipédia relacionadas à pergunta e, em seguida, a utilização dessas informações por um modelo para compor a resposta. Entretanto, o termo *Retrieval-Augmented Generation* surgiu apenas em 2021, com o artigo intitulado *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (Lewis et al. 2020). Nesse trabalho, os autores apresentam uma proposta voltada a tarefas que exigem acesso a conteúdos especializados. Como nem todas as informações cabem na entrada do modelo, a solução passa a incluir apenas os trechos mais relacionados à consulta. Esses fragmentos compõem o contexto usado na geração, o que aproxima a resposta da pergunta original e reduz o risco de construções imprecisas (Huyen 2025).

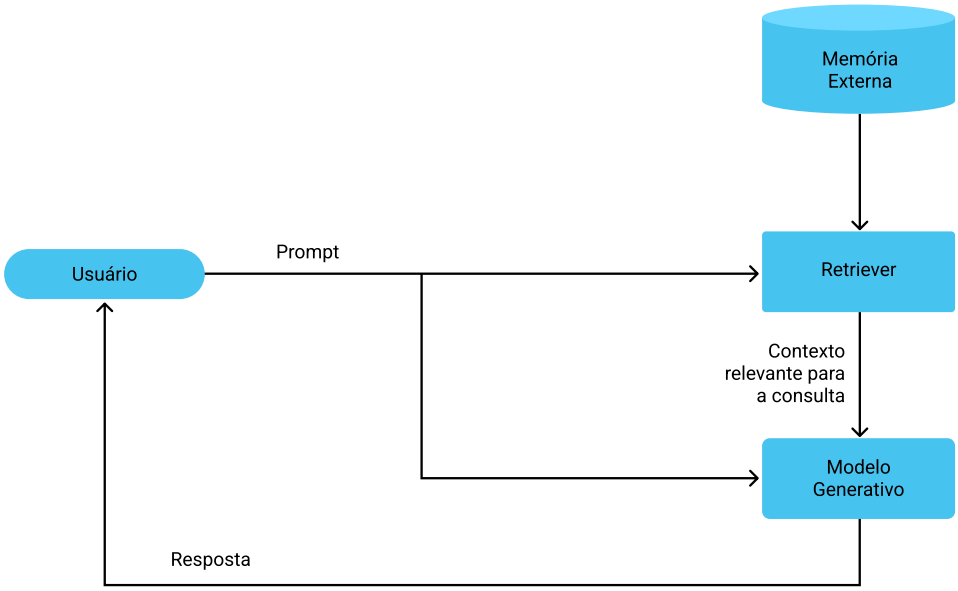
O funcionamento do RAG pode ser dividido em duas etapas principais (Gao et al. 2023, Huyen 2025). Na primeira, ocorre a recuperação de informações a partir de fontes de conhecimento externas, permitindo acesso a dados atualizados e específicos. Isso reduz o risco de respostas incorretas, conhecidas como "alucinações", e oferece uma alternativa eficiente em custo quando comparada a métodos como o *fine-tuning* (Huyen 2025).

Na etapa seguinte, as informações recuperadas são integradas ao *prompt* utilizado pelo modelo. A consulta inicial é processada pelo sistema de recuperação, que localiza dados relevantes. Esses dados são, então, incorporados ao *prompt* enviado ao modelo de linguagem. Com o contexto enriquecido, o modelo gera uma resposta que combina seu conhecimento prévio com os dados específicos recuperados, como ilustra a Figura 2.7 (Huyen 2025).

Agentic Retrieval-Augmented Generation

Conforme Li, Wang, Wang, Hung, Xie & Wang (2025), o *Agentic Retrieval-Augmented Generation* (*Agentic* RAG) é uma evolução da técnica de *Retrieval-Augmented Genera-*

Figura 2.7: Fluxo geral de um RAG.



Fonte: Adaptada de Huyen (2025).

tion, incorporando agentes de IA para superar limitações das *pipelines* tradicionais. O RAG tradicional combina um módulo de recuperação de informações, que utiliza um modelo de *embeddings* e uma base de dados vetorial para encontrar documentos relevantes, com um modelo generativo que usa o contexto recuperado para produzir respostas. Embora eficiente, essa abordagem apresenta restrições, como a ausência de mecanismos de validação sobre a qualidade das informações recuperadas.

O *Agentic* RAG tenta resolver essas limitações ao integrar agentes capazes de orquestrar os componentes da *pipeline* e realizar ações adicionais. Esses agentes podem combinar capacidades de memória, planejamento e uso de ferramentas externas. Por exemplo, eles podem decidir qual fonte de informação consultar, formular consultas otimizadas e iterar sobre as buscas para refinar as respostas geradas (Li, Wang, Wang, Hung, Xie & Wang 2025).

Essa abordagem permite que o sistema lide com cenários mais dinâmicos e complexos. Enquanto um RAG tradicional realiza apenas uma rodada de recuperação e geração, o *Agentic* RAG utiliza iterações para ajustar consultas e verificar a relevância do contexto recuperado até que uma resposta satisfatória seja produzida. Essa evolução torna o *Agentic* RAG uma solução robusta e versátil para aplicações baseadas em LLMs, como assistentes virtuais, fluxos de trabalho automatizados e sistemas de suporte técnico (Li, Wang, Wang, Hung, Xie & Wang 2025).

Seção 3

Trabalhos Relacionados

As investigações que articulam o uso da técnica RAG com estratégias de Engenharia de *Prompts* no setor automotivo ainda são limitadas em escopo e quantidade (Gao et al. 2023, Li, Li, Yang, Yang, Chi & Yang 2025). Apesar do avanço das aplicações baseadas em modelos de linguagem, observa-se um número reduzido de estudos que exploram a recuperação de informação em contextos técnicos específicos, como o de manuais veiculares. A seguir, são apresentados alguns trabalhos que oferecem subsídios comparativos e ajudam a contextualizar a proposta desenvolvida neste estudo.

No trabalho de Mao et al. (2023), é proposta uma mudança de paradigma no desenvolvimento de sistemas de condução autônoma. Ao integrar Modelos de Linguagem de Grande Escala como um agente cognitivo, busca-se incorporar inteligência semelhante à humana nos sistemas autônomos de direção. Este agente cognitivo, denominado Agent-Driver, é capaz de realizar raciocínio em cadeia de pensamentos, planejamento de tarefas, planejamento de movimentos e auto-reflexão. Os resultados indicam que o Agent-Driver supera de forma expressiva os métodos tradicionais, abrindo caminho para abordagens com características mais próximas do comportamento humano na condução de veículos.

Já no estudo de Amato et al. (2023), explora-se a criação do CREA2, um agente conversacional inteligente voltado para o domínio jurídico. Esse agente é capaz de guiar usuários na compreensão de procedimentos e termos jurídicos, além de auxiliar na elaboração de documentos legais. Com um foco especial em resolver disputas na União Europeia, o CREA2 emprega algoritmos avançados de Processamento de Linguagem Natural e o modelo *Sentence Bidirectional Encoder Representations from Transformers* (SBERT) para avaliar a similaridade entre as consultas dos usuários e as questões de sua base de conhecimento. O estudo avaliou dois modelos SBERT pré-treinados, destacando-se o quora-distilbert-multilingual que, após o *fine-tuning*, apresentou melhoria no F1-score, evidenciando desempenho superior na identificação de questões jurídicas semelhantes. Apesar de desafios como o tamanho limitado do conjunto de dados, os resultados indicam o potencial desses modelos na construção de um sistema de recuperação de informações úteis para a resolução de conflitos legais.

No campo automotivo, Medeiros et al. (2023) conduziram um estudo comparativo para avaliar o desempenho de três abordagens de *chatbots* baseadas em IA no processamento de manuais automotivos em formato PDF. O estudo emprega conceitos atualmente associados ao RAG, que combina LLMs com mecanismos de indexação vetorial e recuperação semântica, por meio de ferramentas como LlamaIndex e LangChain, além da

biblioteca Sentence Transformers, utilizada para a geração de *embeddings*. Para realizar a análise, o estudo utilizou como caso prático o manual do Ford Fiesta 2015, avaliando precisão, custo e experiência do usuário, e identificou características distintas entre as abordagens. Nesse contexto, uma das abordagens, que envolvia interação direta via código, mostrou-se moderadamente precisa, mas necessitava de uma interface. Outra abordagem, que utilizava uma interface baseada em Streamlit, apresentou a melhor relação custo-precisão, embora tenha enfrentado dificuldades ao lidar com a interpretação de símbolos e ícones. Por fim, uma terceira abordagem, com *front-end* em React e *back-end* com FastAPI e Qdrant, mostrou-se mais econômica, mas apresentou menor precisão nas respostas. Desse modo, o estudo conclui que a escolha da solução de *chatbot* mais adequada para interagir com manuais automotivos está atrelada aos critérios priorizados pelo usuário. Entre esses critérios, incluem-se a precisão das informações, o custo de utilização e a qualidade da experiência de interação. Diante desse cenário, evidencia-se o potencial de abordagens que combinam recuperação de informação e modelos generativos para otimizar e facilitar o acesso a dados técnicos.

Por outro lado, Siddharth & Luo (2024) desenvolveram um método para aprimorar sistemas de RAG no contexto de projeto e análise de engenharia, com foco na extração de informações estruturadas a partir de descrições de patentes. O procedimento identifica relações explícitas do tipo *head entity* :: *relation* :: *tail entity*, extraindo dados sobre a estrutura e a funcionalidade de artefatos. Para isso, os autores compilaram um conjunto de 375.084 exemplos (positivos e negativos) oriundos de 44.227 sentenças e realizaram *fine-tuning* de modelos de linguagem em duas tarefas: (1) classificação de *tokens*, que atingiu 99,7% de acurácia, e (2) geração Seq2Seq, que elicita diretamente a relação. Aplicado a 4.870 patentes de sistemas de ventilação (classe CPC F04D), o método resultou em uma base de conhecimento com cerca de 2,93 milhões de triplas (*head entity – relation – tail entity*), usada para orientar LLMs na redação de respostas técnicas coesas e precisas.

Além disso, Chandrasekhar et al. (2024) apresentaram o *AMGPT*, um modelo de linguagem voltado a consultas contextuais em manufatura aditiva, baseado em RAG. O sistema parte do LLaMA 2-7B e integra conhecimento extraído de cerca de 50 artigos e livros sobre manufatura aditiva, os quais foram convertidos para o formato *TeX* com uso do Mathpix e posteriormente indexados com o LlamaIndex, sendo a orquestração realizada por meio do LangChain. Adicionalmente, emprega a *ScienceDirect Search* API para recuperar publicações em tempo real, o que possibilita a geração de respostas atualizadas e alinhadas ao estado da arte. Para garantir equilíbrio entre precisão e criatividade, parâmetros como o *top-k* de similaridade e a temperatura de amostragem foram ajustados de forma criteriosa. Em avaliação conduzida por especialistas, o AMGPT produziu respostas factuais sem alucinações em 80% dos *prompts* (contra 86,7% do GPT-4), demonstrando maior especificidade técnica e coerência no domínio da manufatura aditiva.

De forma complementar, Wang et al. (2024) conduziram um estudo de viabilidade para avaliar o uso de LLMs em tarefas de codificação de registros de câncer em ambiente hospitalar. Para isso, o sistema proposto adota uma abordagem baseada em RAG, utilizando o modelo Mistral-7B em combinação com *prompts* elaborados por meio de técnicas de *prompt engineering*. Entre os recursos incorporados, destaca-se o uso de raciocínio em cadeia (*Chain-of-Thought*, CoT), que contribuiu para aprimorar a consistência e a com-

pletude dos códigos gerados. Com isso, o sistema alcançou um F1-score macro-médio de 0,824 com *prompts* detalhados e raciocínio estruturado, superando em 0,187 pontos a configuração sem CoT (F1 = 0,637).

Adicionalmente, Woo et al. (2024) conduziram um estudo comparativo para avaliar o desempenho de modelos de linguagem baseados em RAG e agentes de IA no manejo de informações médicas, utilizando como estudo de caso o tratamento de lesões no Ligamento Cruzado Anterior (ACL). A pesquisa incluiu tanto modelos de código fechado, como GPT-4 e Claude 3, quanto modelos de código aberto, como LLaMA 3-70B e Mistral-7B. Inicialmente, todos os modelos apresentaram desempenho inferior a 60% em sua configuração base, sendo o Claude 3 o mais elevado (58%) e o GPT-3.5 o mais baixo (44%). Para mitigar essas limitações, os autores implementaram uma abordagem RAG fundamentada nas diretrizes clínicas da *American Academy of Orthopaedic Surgeons* (AAOS) para o manejo de ACL. Com isso, observou-se um aumento médio de 39,7 pontos no F1-score, com destaque para o LLaMA 3-70B, que alcançou 94%. Além disso, ao incorporar agentes de IA (*Agentic* RAG), o GPT-4 obteve 95% de F1-score, indicando o potencial dessa estratégia para refinar ainda mais as respostas. O *Agentic* RAG integra múltiplos modelos com funções especializadas, como validação de qualidade e iteração de respostas, promovendo maior alinhamento com as diretrizes da AAOS. Por fim, as métricas ROUGE (1/2/L) e METEOR foram utilizadas para avaliar a similaridade semântica e lexical, sendo que o GPT-4 com *Agentic* RAG obteve os melhores resultados: ROUGE-1 de 0,59 e METEOR de 0,50.

Aribas & Daglarli (2024) investigaram, por sua vez, a integração de modelos de personalidade com técnicas avançadas de IA Generativa, em particular a técnica de RAG, para aprimorar sistemas de recomendação personalizada de viagens. O estudo combinou LLMs com modelos de personalidade, como o Indicador de Tipos de Myers-Briggs (MBTI) e os traços de personalidade Big Five (BF), viabilizando recomendações mais contextualizadas e ajustadas às preferências individuais. Por meio de um sistema que processa consultas em linguagem natural, realiza recuperação semântica e gera recomendações personalizadas, os resultados demonstraram índices de precisão de 82%, satisfação do usuário de 78% e sucesso na adequação das recomendações a traços de personalidade, como extroversão (85%) e introversão (75%).

Por fim, Heredia Álvaro & Barreda (2025) apresentaram um sistema avançado de RAG especializado para controle de qualidade no processo de fabricação de azulejos cerâmicos. Diferentemente de um RAG genérico, o sistema incorpora mecanismos como pré-processamento e pós-processamento personalizados, além de reordenação (*reranking*) de informações recuperadas. Na avaliação da recuperação, foram analisados parâmetros como o número de amostras relevantes (k) e o limiar para filtragem, utilizando métricas como *Jaccard Similarity*, *F1-Score* e *Kendall-Tau distance* para medir a precisão e a relevância na identificação das informações. Já na geração, a métrica *ROUGE-L* foi utilizada para avaliar a qualidade das respostas em termos de coerência e fidelidade ao conteúdo esperado. Com isso, como resultados, o sistema demonstrou alta eficiência na recuperação de informações relevantes, com uma *Jaccard Similarity* de 92.68% e um *F1-Score* de 85.81%, e apresentou um *ROUGE-L* médio de 0,61 na geração de respostas.

Assim, embora diversos estudos explorem o uso de LLMs em áreas como saúde, ma-

nufatura, direito e personalização de serviços, conforme apresentados na 3.1, muitos não abordam diretamente o RAG, enquanto outros se limitam a aplicações básicas dessa técnica, sem explorar métodos mais avançados, como *Agentic RAG*, por exemplo. O trabalho de Mao et al. (2023) utiliza agentes cognitivos para direção autônoma, enquanto Amato et al. (2023) aplica SBERT em consultas jurídicas, ambos sem empregar RAG diretamente. Por outro lado, Heredia Álvaro & Barreda (2025) utiliza RAG para controle de qualidade na fabricação de azulejos cerâmicos, e Siddharth & Luo (2024) explora a extração de conhecimento estruturado, mas sem se aprofundar em técnicas avançadas relacionadas ao RAG.

Ademais, Chandrasekhar et al. (2024) e Wang et al. (2024) aplicam RAG em manufatura aditiva e codificação de registros de câncer, respectivamente, com contribuições relevantes em consultas contextuais e melhorias de precisão. No campo da saúde, Woo et al. (2024) apresenta o *Agentic RAG* para manejo clínico, enquanto Medeiros et al. (2023) utiliza RAG no processamento de manuais automotivos, destacando-se como um dos poucos estudos com foco direto no setor automotivo. Já Aribas & Daglarli (2024) emprega RAG em sistemas de recomendação baseados em modelos de personalidade. Apesar dessas contribuições, a integração de RAG com Engenharia de *Prompts* e outras abordagens avançadas ainda representa um campo promissor e pouco explorado, especialmente no setor automotivo.

Dessa forma, este estudo propõe investigar a aplicação da técnica de RAG no setor automotivo em articulação com métodos voltados à organização de fluxos e à estruturação de consultas. Para isso, são incorporadas práticas de Engenharia de *Prompts*, incluindo estratégias de refinamento com base em Gradiente Descendente, bem como mecanismos de orquestração de agentes, como o *Agentic* RAG. A proposta contempla ainda a análise integrada de diferentes variantes de recuperação e geração, em um ambiente orientado por LLMs. Com essa abordagem, busca-se contribuir para o avanço das soluções voltadas à consulta e ao processamento de manuais técnicos, especialmente em contextos que envolvem assistência ao usuário em cenários veiculares.

Tabela 3.1: Resumo dos trabalhos relacionados

Trabalho \ Características	Arquitetura Utilizada	Tecnologias Empregadas	Aplicações	Validação	Resultados
Mao et al. (2023)	Agente cognitivo (*Agent-Driver*)	LLMs, raciocínio em cadeia de pensamentos (*CoT*)	Condução autônoma com inteligência humanizada	Testes com benchmarks de direção autônoma	Alta eficiência em sistemas autônomos
Amato et al. (2023)	Modelo SBERT com *fine-tuning*	Processamento de Linguagem Natural (PLN), similaridade semântica	Assistência legal e resolução de disputas	Testes em sistemas jurídicos da União Europeia	Melhor *F1-score* em consultas jurídicas
Medeiros et al. (2023)	RAG para processamento de manuais automotivos	LlamaIndex, LangChain, Sentence Transformers	Interação com manuais em formato PDF	Avaliação com precisão, custo e experiência do usuário, utilizando *prompts zero-shot*, *one-shot* e *few-shot*	Melhor relação custo-precisão (Streamlit); Mais econômico, menor precisão (React/FastAPI).
Siddharth & Luo (2024)	Extração de conhecimento estruturado	*Fine-tuning* de LLMs para classificação de tokens e Seq2Seq	Construção de bases de conhecimento para auxiliar LLMs em geração técnica	Avaliação técnica com tarefas de recuperação e geração	Precisão de até 99,7% na identificação de relações; 2,93 milhões de fatos extraídos
Chandrasekhar et al. (2024)	RAG para manufatura aditiva	LLaMA2-7B, Mathpix, LlamaIndex	Consultas contextuais sobre propriedades de materiais e processos de manufatura aditiva	Validação comparativa e ajuste de parâmetros como *top-k similarity* e temperatura	Respostas factuais em 80% das consultas; maior especificidade em relação ao GPT-4
Wang et al. (2024)	RAG para codificação de registros de câncer	Mistral-7B, *prompt engineering* com raciocínio em cadeia (*CoT*)	Codificação automatizada de registros de câncer em hospitais	Avaliação com tarefas específicas de registro	*F1-score* macro-médio de 0,824; ganho de 0,187 pontos com *CoT*
Woo et al. (2024)	RAG com *Agentic RAG* para manejo médico	GPT-4, Llama3-70b, Claude3, Mistral-7b	Diretrizes clínicas, como manejo de lesões do Ligamento Cruzado Anterior	Avaliação com F1-score, ROUGE e METEOR	Aumento médio de 39,7 pontos no F1-score; GPT-4 com *Agentic RAG* atingiu 95% de F1-score
Aribas & Daglarli (2024)	RAG aplicado a sistemas de recomendação baseados em modelos de personalidade	MBTI, Big Five	Recomendação personalizada de viagens com base em traços de personalidade	Avaliação com métricas de precisão, satisfação do usuário e adequação às características de personalidade	Precisão de 82%, satisfação de 78%, adequação de 85% para extroversão e 75% para introversão
Heredia Álvaro & Barreda (2025)	RAG especializado com pré e pós-processamento	LLMs com reordenação (*reranking*)	Controle de qualidade na fabricação de azulejos cerâmicos	Avaliação com *Jaccard*, *F1*, *Kendall-Tau*, *ROUGE-L* e validação prática	*Jaccard Similarity* de 92.68%; *F1-Score* de 85.81%; *ROUGE-L* médio de 0,61

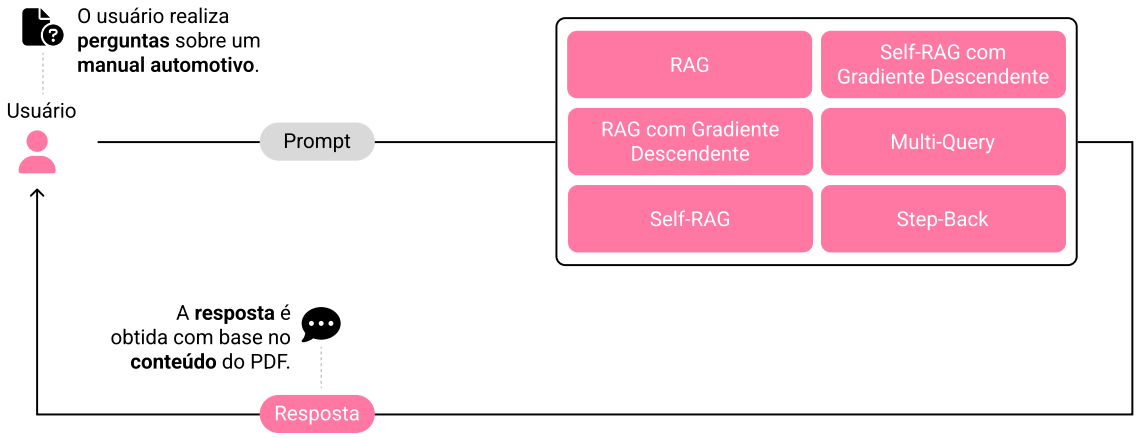
Fonte: Elaborada pela autora (2025).

Seção 4

Abordagem Proposta

Nesta seção, apresenta-se a abordagem proposta para lidar com a extensão e a complexidade dos manuais técnicos, fatores que dificultam a obtenção de informações específicas pelos usuários. Com o intuito de superar esse desafio, foram desenvolvidas seis variantes baseadas na técnica RAG, conforme ilustrado na Figura 4.1, cada uma fundamentada em estratégias distintas de otimização e adaptabilidade. Essas variantes são detalhadas a seguir.

Figura 4.1: Fluxo geral das abordagens baseadas na técnica RAG.



Fonte: Elaborada pela autora (2025).

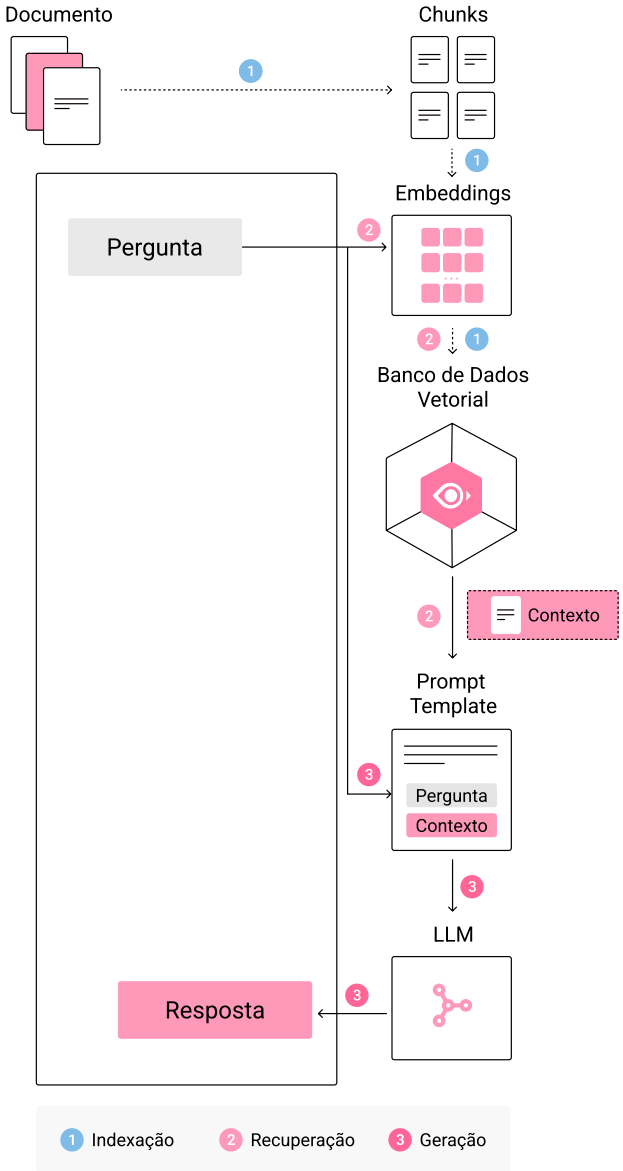
4.1 *Retrieval-Augmented Generation*

O RAG desenvolvido neste estudo, aplicado ao contexto de manuais veiculares, foi estruturado com base em etapas bem definidas. A abordagem busca integrar recuperação de informações com geração contextualizada de respostas. O processo é dividido em três partes principais, conforme descrito a seguir e representado na Figura 4.2.

1. **Indexação** — Inicialmente, os documentos técnicos são convertidos para um formato textual estruturado, preservando seções, títulos e demais elementos relevantes.

Após essa conversão, cada arquivo é segmentado em partes menores, denominadas *chunks*, conforme critérios de tamanho e organização do conteúdo. Em cada segmento, são incluídos metadados como marca, modelo e ano, extraídos do nome do arquivo. Por fim, cada *chunk* é transformado em um vetor de *embeddings*, representando seu conteúdo semântico. Todos os vetores são armazenados em um banco de dados vetorial, possibilitando a recuperação por similaridade nas etapas seguintes.

Figura 4.2: Representação do processo RAG.



Fonte: Adaptada de Cardenas & Monigatti (2024).

2. **Recuperação** — Após a indexação, procede-se ao processamento da consulta fornecida pelo usuário, com o objetivo de identificar no índice os fragmentos mais relevantes. O usuário informa a marca, o modelo e o ano do veículo, possibilitando

uma filtragem inicial que restringe a busca ao conjunto de documentos pertinente. Posteriormente, a consulta é convertida em um vetor de *embeddings*, o qual é comparado aos vetores previamente armazenados. Como resultado desse processo, os fragmentos recuperados são ordenados de acordo com sua similaridade semântica. Assim, os mais relevantes são selecionados para a etapa seguinte.

3. **Geração** — Na etapa final, os fragmentos recuperados são integrados à consulta fornecida pelo usuário, compondo o contexto de entrada para o modelo gerador. Por meio de um modelo de linguagem, mais especificamente um *LLM*, o sistema produz uma resposta textual que combina as informações recuperadas com a intenção expressa na consulta.

Este processo sequencial define, portanto, o princípio geral da técnica RAG: a produção de respostas fundamentadas em informações extraídas diretamente da fonte documental. A partir dessa base, estruturam-se as extensões da abordagem, as quais buscam aprimorar a adequação do contexto por meio de refinamento iterativo e de critérios de decisão adaptativos.

4.2 *Retrieval-Augmented Generation* com Gradiente Descendente

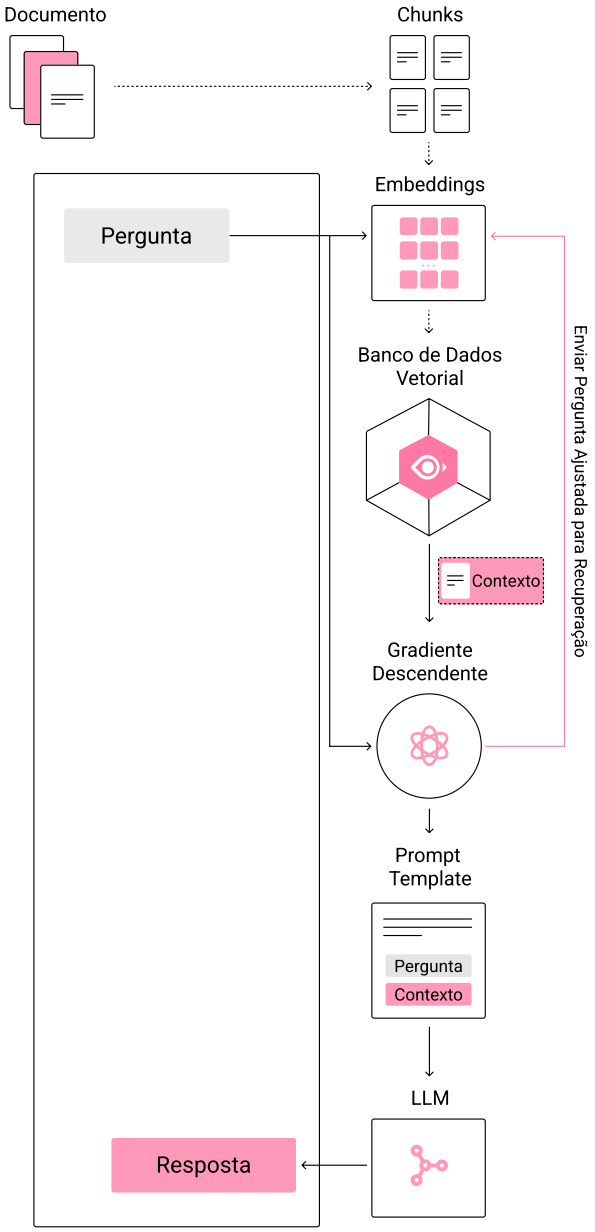
Nesta abordagem, o RAG é estendido para incluir um mecanismo iterativo baseado em Gradiente Descendente, que valida a resposta só após superar um limiar de utilidade, reformulando a consulta quando isso não ocorre, visando otimizar a recuperação e a geração de respostas. O fluxo segue as etapas descritas anteriormente, que incluem a preparação e indexação dos documentos, a consulta do usuário e recuperação dos fragmentos relevantes e, por fim, a geração de respostas com base nos fragmentos recuperados, conforme ilustrado na Figura 4.3.

Inicialmente, o processo é iniciado a partir da consulta fornecida pelo usuário, composta por informações como marca, modelo, ano do veículo e uma pergunta específica. Com base nessa consulta, realiza-se uma busca no banco de dados vetorial, a partir da qual são recuperados documentos considerados relevantes. Esses documentos são segmentados em fragmentos textuais, que passam a compor o contexto inicial utilizado para a geração da resposta.

Em seguida, o modelo de linguagem gera uma primeira resposta com base no contexto recuperado. Posteriormente, este resultado é submetido à etapa de avaliação automática, que verifica se a resposta efetivamente resolve a questão fornecida e atribui um nível de utilidade em uma escala de 1 a 5. Para isso, a avaliação leva em consideração fatores como clareza, completude e pertinência das informações. Caso a resposta obtenha utilidade inferior ao patamar definido (utilidade menor que 4) ou não seja suficiente para resolver a questão, o sistema identifica pontos de melhoria na formulação da pergunta, como falta de especificidade, uso de termos genéricos ou ausência de detalhes relevantes. A partir desse diagnóstico, são extraídos motivos concretos para o desempenho insatisfatório, processo

análogo ao cálculo de gradientes em métodos de otimização. Esses "gradientes semânticos" orientam a reformulação da consulta original, resultando em uma versão revisada da pergunta, que busca direcionar de forma mais precisa a recuperação e a geração de respostas.

Figura 4.3: Representação do processo RAG utilizando Gradiente Descendente.



Fonte: Adaptada de Cardenas & Monigatti (2024).

Com a pergunta reformulada, o ciclo de busca e geração é repetido, envolvendo nova recuperação de fragmentos, atualização do contexto e produção de uma resposta revisada. O processo é interrompido assim que uma resposta atinge o critério de utilidade estabelecido ou quando se completa o número máximo de iterações previsto no fluxo. Dessa

forma, a abordagem emprega um ajuste iterativo guiado por *feedback*, promovendo refinamento contínuo do contexto e da consulta até a obtenção de uma resposta mais precisa e alinhada à intenção do usuário.

4.3 *Self-Reflective Retrieval-Augmented Generation* Adaptativo

Como extensão do RAG, o *Self-RAG* Adaptativo introduz melhorias no processo de recuperação e geração ao ajustar dinamicamente as informações recuperadas e a forma como são utilizadas no contexto, como representado na Figura 4.4. O fluxo deste processo pode ser estruturado e representado como um grafo direcionado, no qual cada nó corresponde a uma etapa ou decisão específica, e as arestas indicam os caminhos possíveis entre essas etapas. Conforme descrito por Asai et al. (2023), essa abordagem incorpora elementos de *Self-Reflection*, permitindo que cada etapa seja avaliada para identificar possíveis falhas e propor reformulações, característica importante em manuais automotivos, pois perguntas podem exigir ajustes sucessivos de contexto para cobrir seções dispersas ou termos não padronizados.

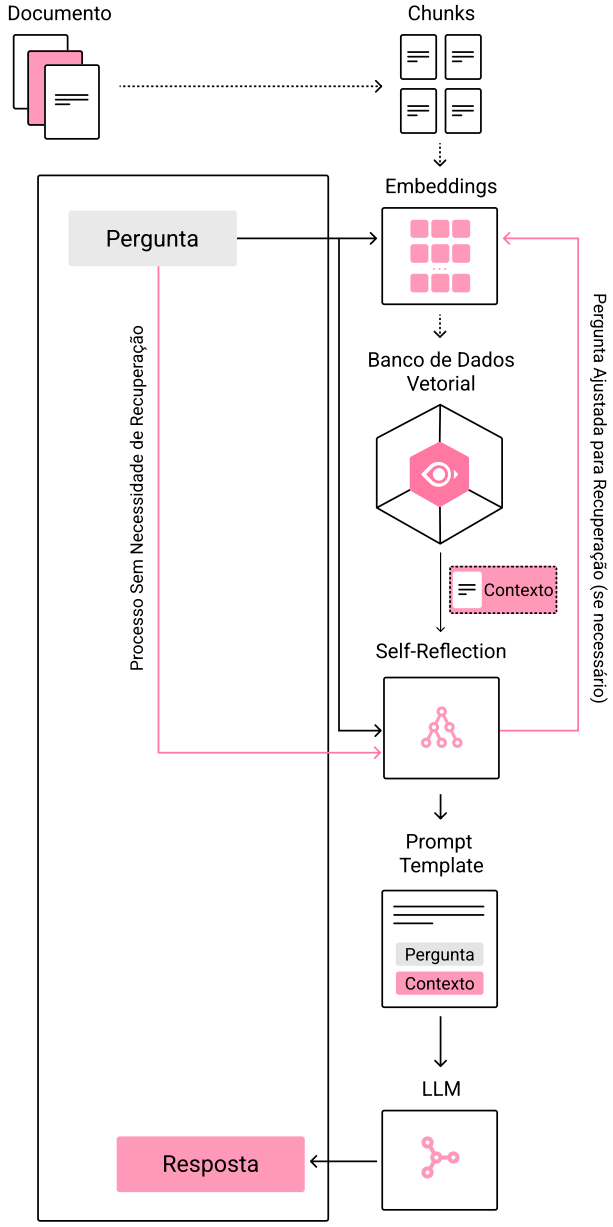
Inicialmente, a consulta do usuário, composta por informações como marca, modelo, ano do veículo e a pergunta de interesse, é analisada por um mecanismo automático que avalia se há necessidade de recuperação de documentos (*Retrieval*), se a resposta pode ser fornecida diretamente sem busca (*No Retrieval*) ou se é possível prosseguir utilizando as evidências previamente disponíveis (*Continue to Use Evidence*). Caso a decisão aponte para a necessidade de recuperação, verifica-se a existência de dados correspondentes no banco vetorial. Não havendo informações, busca-se um arquivo local de manual para ingestão automática. Caso esse procedimento também não seja viável, o fluxo é encerrado com a comunicação da ausência de dados.

Em seguida, na etapa de recuperação, os documentos potencialmente relevantes são identificados com base nos filtros aplicados (marca, modelo, ano e conteúdo da consulta). Cada fragmento recuperado é, então, avaliado automaticamente pelo modelo de linguagem, que utiliza o *token* IsREL para determinar se o conteúdo é relevante para a pergunta. Fragmentos considerados irrelevantes são descartados, enquanto aqueles classificados como relevantes formam o conjunto de evidências que será empregado nas etapas seguintes. Dessa forma, garante-se que apenas informações pertinentes à demanda do usuário integrem o contexto de geração, minimizando conteúdos dispersos ou pouco relacionados à resposta final.

Em sequência, o modelo de linguagem gera uma resposta fundamentada nos fragmentos relevantes. Essa resposta é avaliada quanto ao seu grau de suporte factual, por meio do *token* IsSUP, que distingue entre respostas totalmente suportadas, parcialmente suportadas ou contraditórias em relação à base consultada. Sempre que a resposta não apresenta fundamentação adequada ou contém contradições, o que caracteriza a presença de alucinação, o fluxo aciona o mecanismo de reescrita da consulta, buscando reduzir a ocorrência de informações infundadas.

Além da verificação da fundamentação, o fluxo inclui a avaliação da resposta em dois

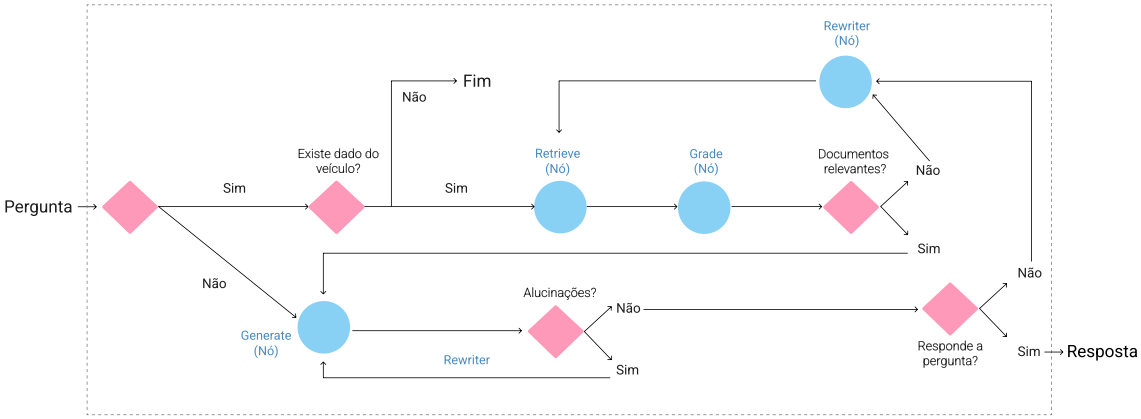
Figura 4.4: Representação do processo *Self-RAG* Adaptativo.



Fonte: Adaptada de Cardenas & Monigatti (2024).

aspectos complementares. Primeiramente, o sistema utiliza um critério binário para determinar se a resposta, de fato, resolve a pergunta do usuário, classificando-a como “sim” ou “não”. Em seguida, atribui-se um nível de utilidade, por meio do *token* IsUSE, que varia de 1 a 5, refletindo o grau de completude, clareza e adequação da resposta. Sempre que a resposta não alcança o padrão mínimo esperado nesses critérios, a consulta original é reformulada, com base em orientações extraídas do próprio processo de avaliação, com o intuito de torná-la mais alinhada ao conteúdo técnico dos manuais. Esse ciclo de reescrita e reavaliação pode ocorrer mais de uma vez, porém respeita um limite de tentativas, de modo a evitar repetições indefinidas. Assim, caso esse máximo seja atingido sem sucesso, o processo se encerra com a devida sinalização da limitação dos dados disponíveis.

Figura 4.5: Fluxo completo do processo *Self-RAG* Adaptativo.



Fonte: Adaptada de LangChain (2024).

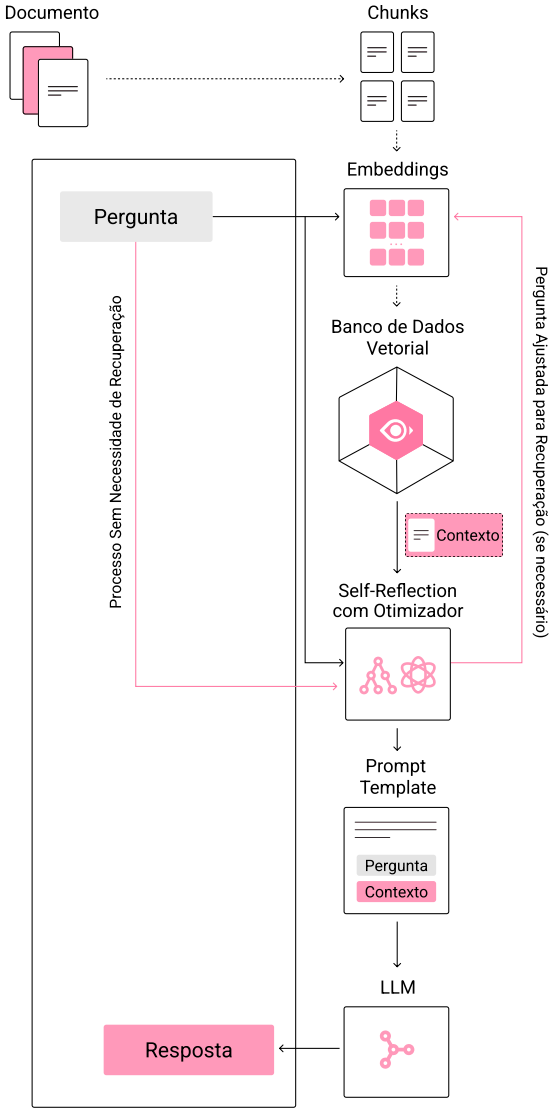
Dessa forma, entre as etapas, decisões condicionais orientam os caminhos possíveis. Por exemplo, quando não há documentos relevantes identificados pelo *token* IsREL, o fluxo direciona para uma nova tentativa de reformulação da consulta. Se a resposta gerada não apresenta fundamentação adequada segundo o *token* IsSUP ou não atinge a utilidade mínima pelo IsUSE, o sistema reinicia o ciclo. Por outro lado, ao identificar uma resposta satisfatória em todos os critérios, o fluxo é finalizado. Assim, o grafo (Figura 4.5) reflete o funcionamento adaptativo e iterativo da abordagem, que busca elevar a qualidade e a pertinência das respostas geradas.

4.4 *Self-Reflective Retrieval-Augmented Generation* Adaptativo com Gradiente Descendente

O *Self-RAG* Adaptativo com Gradiente Descendente preserva as características do *Self-RAG* Adaptativo, como ilustrado na Figura 4.6. No entanto, o Gradiente Descendente é acionado somente quando a avaliação indica que a resposta ou o processo de recuperação não atende aos critérios definidos ao longo do fluxo. Nessas circunstâncias, o sistema utiliza o Gradiente Descendente para revisar a consulta ou ajustar as interações seguintes,

com o objetivo de possibilitar maior alinhamento entre a demanda do usuário e o conteúdo disponível, sem recorrer a intervenções desnecessárias.

Figura 4.6: Representação do processo *Self-RAG* Adaptativo com Gradiente Descendente.



Fonte: Adaptada de Cardenas & Monigatti (2024).

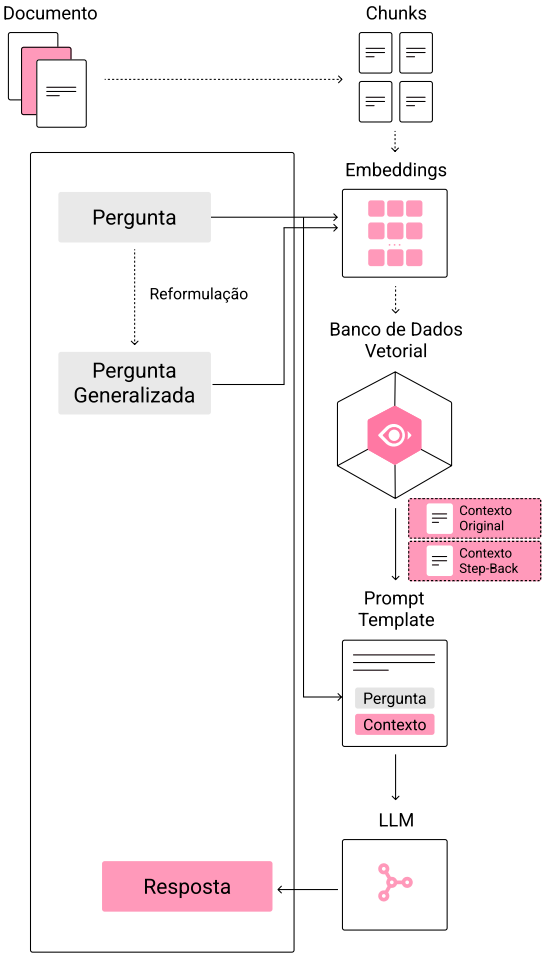
A principal distinção em relação ao fluxo original está na reformulação da consulta. Quando a resposta apresenta limitações de fundamentação, relevância ou utilidade, o otimizador fundamentado em Gradiente Descendente identifica os fatores que levaram ao insucesso e, a partir desse diagnóstico, propõe uma nova versão da consulta, fundamentada em orientações extraídas do próprio processo de avaliação. O procedimento tem como objetivo tornar a consulta mais clara, específica e compatível com a estrutura dos documentos técnicos, o que aumenta a chance de recuperação de fragmentos pertinentes à necessidade apresentada.

Portanto, o emprego do Gradiente Descendente se restringe a situações nas quais são identificadas oportunidades concretas de aprimoramento, como a ampliação da recuperação de informações ou o reforço da fundamentação da resposta. Com isso, a abordagem mantém intervenções focadas e evita a adoção de procedimentos desnecessários ao longo do ciclo adaptativo.

4.5 *Step-Back*

De acordo com Zheng et al. (2024), o *Step-Back* recorre à reformulação da consulta para contornar limitações associadas a perguntas muito específicas ou pouco claras. Como ilustrado na Figura 4.7, esse método amplia o conjunto de evidências avaliadas durante a geração da resposta e favorece a recuperação de informações pertinentes mesmo diante de perguntas iniciais restritivas. No contexto dos manuais automotivos, a reformulação pode viabilizar o acesso a trechos redigidos em termos mais gerais, além de atenuar variações de nomenclatura entre fabricantes.

Figura 4.7: Representação do processo *Step-Back*.



Fonte: Adaptada de Cardenas & Monigatti (2024).

O fluxo inicia-se com o recebimento da consulta do usuário, acompanhada dos filtros marca, modelo e ano do veículo. Na sequência, um modelo de linguagem gera uma versão mais genérica da pergunta, a fim de explicitar o objetivo principal da demanda e facilitar a identificação de fragmentos pertinentes.

Na etapa de recuperação, o sistema realiza duas buscas paralelas e independentes. A primeira utiliza diretamente a formulação original da pergunta, mantendo os termos empregados pelo usuário. A segunda utiliza a versão genérica, gerada pelo modelo, com o objetivo de ampliar o escopo da busca e alcançar trechos do manual que abordem o mesmo assunto com outras palavras.

Os resultados obtidos nas duas buscas são reunidos em um único conjunto de evidências. Nessa etapa, o sistema remove documentos repetidos ou muito semelhantes, a fim de evitar o desperdício de espaço no contexto e reduzir o risco de sobreposição desnecessária de informações. A junção dos trechos selecionados por diferentes perspectivas permite integrar variações na forma de expressão do conteúdo, ampliando as chances de o modelo gerar uma resposta mais informativa e alinhada à demanda.

Finalmente, ambos os contextos, decorrentes da consulta original e da versão generalizada, são utilizados pelo modelo para compor a resposta. Com isso, o procedimento busca combinar a precisão do contexto restrito com a abrangência proporcionada pela versão generalizada, favorecendo respostas mais completas e menos suscetíveis a omissões.

4.6 *Multi-Query*

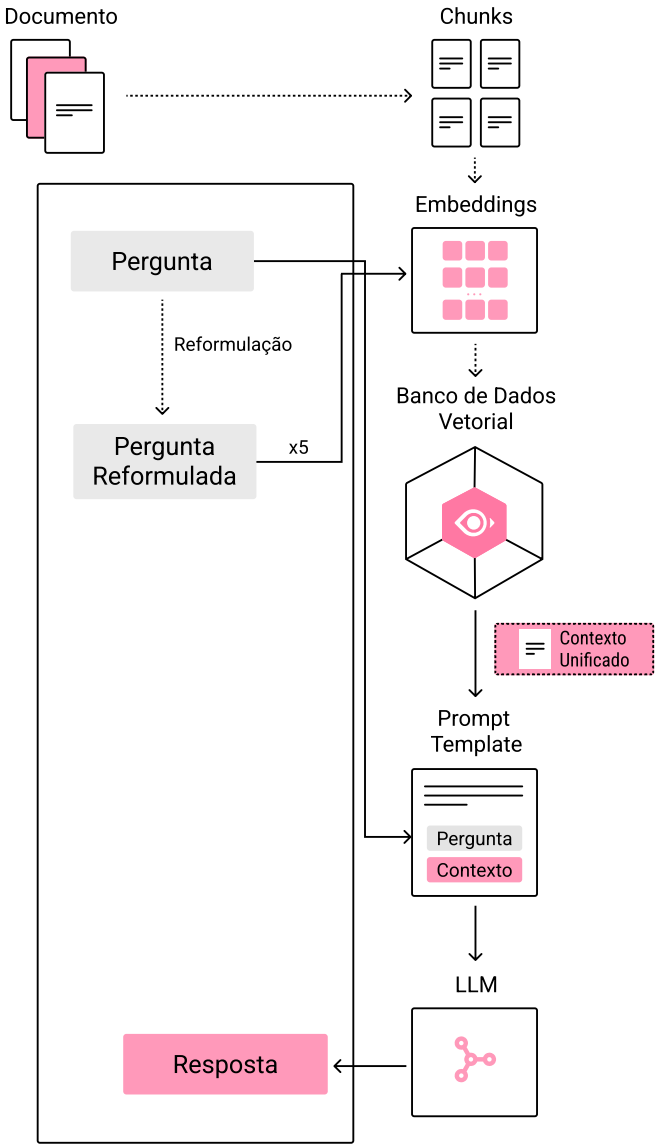
Conforme Kostric & Balog (2024), a técnica *Multi-Query* amplia a etapa de recuperação ao gerar, para cada solicitação do usuário, um conjunto de reformulações semânticas que abordam diferentes perspectivas sobre o mesmo tema. No contexto dos manuais automotivos, essa estratégia mostra-se útil porque um mesmo componente pode aparecer sob denominações variadas entre marcas ou até entre seções do manual. Ao produzir perguntas alternativas, o sistema aumenta a probabilidade de capturar passagens que empreguem terminologias distintas, sem exigir que o usuário antecipe todas as variações possíveis. A Figura 4.8 apresenta esse fluxo.

Em primeiro lugar, a consulta do usuário é acompanhada por filtros relacionados à marca, ao modelo e ao ano do veículo. Em seguida, um modelo de linguagem recebe a pergunta original e, a partir de um *prompt* específico, elabora cinco reformulações distintas. Cada variação procura expressar a demanda inicial por meio de sinônimos, construções mais amplas ou abordagens complementares, com o intuito de contornar limitações da recuperação baseada unicamente em similaridade vetorial.

Na etapa seguinte, cada reformulação é utilizada em buscas separadas, que resultam em trechos documentais potencialmente relevantes. Esses fragmentos são, então, reunidos em um único conjunto, no qual duplicações são removidas.

Posteriormente, apenas a formulação original da pergunta, acompanhada dos trechos recuperados, é integrada ao *prompt* de resposta. As reformulações anteriores, por sua vez, têm função auxiliar na recuperação e não são reaproveitadas na construção textual. Nessa fase, o modelo é instruído a redigir respostas com até cinco frases, a indicar com exatidão

Figura 4.8: Representação do processo *Multi-Query*.



Fonte: Adaptada de Cardenas & Monigatti (2024).

a localização de componentes mencionados e a evitar conjecturas quando a informação solicitada estiver ausente do material consultado.

Dessa forma, a técnica *Multi-Query* amplia o escopo da recuperação por meio de reformulações automatizadas da consulta, mantendo os filtros de metadados e evitando interferências na etapa de geração. Com isso, o processo de reformulação permanece restrito ao sistema, sem exigir reformulações manuais por parte do usuário.

Seção 5

Experimento

Nesta seção, são detalhados o experimento conduzido para a avaliação das variantes de RAG no contexto de manuais automotivos e a classificação metodológica da pesquisa. Na Seção 5.1, aborda-se a caracterização da abordagem adotada e, nas seções subsequentes, apresenta-se o experimento, estruturado em três etapas principais: Planejamento, Operação e Métricas de Avaliação.

5.1 Classificação da Pesquisa

Para que os objetivos desta pesquisa sejam atingidos de forma clara e estruturada, é necessário definir sua classificação metodológica. Nesse sentido, adota-se uma abordagem de caráter exploratório, justificada pela necessidade de investigar a aplicação de variantes de *Retrieval-Augmented Generation* em um domínio ainda pouco estudado, como a recuperação e a geração de informações técnicas a partir de manuais automotivos.

Além disso, quanto à sua natureza, este trabalho caracteriza-se como uma pesquisa aplicada, uma vez que busca solucionar, na prática, o problema de acesso ao conteúdo técnico complexo presente em manuais automotivos. Para isso, o estudo utiliza abordagens avançadas de recuperação e geração, com o objetivo de facilitar a consulta e a compreensão de grandes volumes de texto. Assim, espera-se que a solução desenvolvida permita que profissionais da área automotiva e demais usuários que dependem desse conhecimento tenham um acesso mais rápido e eficiente às informações necessárias.

Quanto à forma de abordagem dos dados, o estudo adota uma perspectiva mista, combinando elementos qualitativos e quantitativos. As avaliações referentes à fidelidade ao contexto, relevância em relação à pergunta, completude das informações e aspectos de segurança foram realizadas por um modelo de linguagem com base em critérios pré-definidos. Tais critérios, embora qualitativos, foram operacionalizados por escalas numéricas, viabilizando uma análise quantitativa. Complementarmente, foi utilizada a métrica BERTScore, que mensura a similaridade semântica entre respostas.

Já no que diz respeito aos procedimentos técnicos, esta pesquisa adota uma abordagem experimental, fundamentada na implementação e avaliação de variantes de RAG. Para isso, foram desenvolvidas e testadas seis variantes principais: o RAG convencional, o RAG com Gradiente Descendente, o *Multi-Query*, o *Step-Back*, o *Self-RAG* Adaptativo e o *Self-RAG* Adaptativo com Gradiente Descendente.

Por fim, a avaliação foi conduzida em ambiente controlado, o que permitiu comparações entre as variantes implementadas. O procedimento adotado envolveu tanto a aplicação dos critérios mencionados anteriormente quanto o uso complementar do BERTScore, que possibilitou uma comparação objetiva das respostas geradas.

5.2 Planejamento

Esta seção aborda as etapas iniciais do experimento, incluindo a definição do contexto, a seleção de variáveis e artefatos, o *design* do estudo e a instrumentação para implementar as variantes de RAG. Todas as etapas foram planejadas para garantir uma condução sistemática e uma avaliação comparativa das propostas.

1) **Seleção do Contexto**: o contexto do estudo foi definido com base no cenário de uso de manuais automotivos, focando em veículos modernos e suas necessidades de documentação técnica. O setor automotivo foi escolhido devido à sua relevância e complexidade, especialmente no que diz respeito à manutenção, operação e especificações técnicas de veículos. Esse contexto permite avaliar as variantes de RAG em um cenário realista e aplicável, garantindo que os resultados sejam representativos para situações práticas envolvendo documentação técnica.

2) **Seleção de Variáveis**: as variáveis do estudo foram definidas com o objetivo de permitir uma análise comparativa entre as abordagens. A variável independente correspondeu à técnica de RAG utilizada, considerando seis variantes: RAG convencional, RAG com Gradiente Descendente, *Multi-Query*, *Step-Back*, *Self-RAG* Adaptativo e *Self-RAG* Adaptativo com Gradiente Descendente. Por outro lado, as variáveis dependentes corresponderam às métricas utilizadas para avaliar os resultados, obtidas a partir da análise do texto gerado, da pergunta original e dos trechos de contexto utilizados. Essas métricas incluíram fidelidade, relevância, completude, segurança e similaridade semântica entre as respostas, mensurada por meio do BERTScore. Esse conjunto de critérios permitiu uma comparação consistente entre as variantes testadas.

3) **Seleção de Artefatos:** para a condução do estudo, foram selecionados os manuais dos veículos Volkswagen Polo 2025 e Fiat Argo 2023. A escolha desses modelos considerou dois critérios principais. O primeiro foi a atualidade dos veículos, por se tratarem de versões recentes disponíveis no mercado nacional. O segundo foi a acessibilidade dos manuais por meio de fontes oficiais das respectivas fabricantes, o que assegura a confiabilidade e a integridade do conteúdo. Além disso, ambos os documentos apresentam ampla variedade de informações técnicas, instruções operacionais e procedimentos de manutenção, cobrindo situações que vão desde o uso cotidiano até emergências. Essa diversidade permitiu a formulação de perguntas com diferentes níveis de complexidade, tornando os manuais adequados para avaliar o desempenho das variantes de RAG em um cenário representativo e alinhado com as demandas reais do setor automotivo.

4) **Design do Estudo**: o estudo foi estruturado em três etapas: implementação das variantes de RAG, execução das consultas e avaliação dos resultados. Na primeira etapa, foram implementadas seis variantes de RAG: RAG Convencional, RAG com Gradiente Descendente, *Multi-Query*, *Step-Back*, *Self-RAG* Adaptativo e *Self-RAG* Adaptativo com Gradiente Descendente. Em seguida, as consultas foram realizadas em um ambiente con-

trolado, utilizando como base de conhecimento os manuais dos veículos Volkswagen Polo 2025 e Fiat Argo 2023. Por fim, os resultados foram avaliados por meio da abordagem *LLM-as-a-judge*, em que um LLM atuou como avaliador com base em quatro rubricas específicas: fidelidade, relevância, completude e segurança. Além disso, foi utilizado o BERTScore para comparação direta entre pares de respostas geradas, com o objetivo de complementar a análise sob uma perspectiva semântica automática.

5) **Instrumentação**: para assegurar uma implementação funcional, buscou-se identificar ferramentas adequadas para o processamento de texto, integração de modelos de linguagem, criação de fluxos interativos e gerenciamento de dados não estruturados. Além disso, a linguagem de programação utilizada foi o Python[1], selecionado por sua versatilidade e pelo suporte de bibliotecas amplamente empregadas em tarefas de Inteligência Artificial Generativa e Modelos de Linguagem de Grande Escala. Dessa forma, as principais ferramentas e bibliotecas utilizadas neste trabalho são apresentadas a seguir.

a. **LangChain**: foi utilizada para integrar as etapas de consulta, recuperação e geração. Por meio de suas abstrações modulares, como *chains*, *prompts*, *retrievers* e *tools*, a biblioteca permitiu estruturar e organizar as diferentes variantes do RAG, oferecendo suporte à composição e ao encadeamento dos componentes necessários em cada abordagem (Topsakal & Akinci 2023, Mavroudis 2024);

b. **Milvus**: empregado para o armazenamento e a recuperação de *embeddings* em bases de dados vetoriais (Wang et al. 2021). Entre as alternativas disponíveis, o Milvus foi selecionado por oferecer alta escalabilidade, desempenho estável em consultas semânticas (Srivamsi et al. 2023) e suporte nativo a operações de filtragem por metadados, como marca, modelo e ano. Isso se deve à necessidade de combinar buscas rápidas com filtros contextuais, recurso importante no domínio de manuais automotivos. Além disso, a possibilidade de integração com bibliotecas Python utilizadas em Inteligência Artificial contribui para a organização e a manutenção das etapas implementadas (Wang et al. 2021);

c. **LangGraph**: utilizado para controlar o fluxo de execução em todas as variantes desenvolvidas, permitindo a modelagem de etapas e estados por meio de grafos direcionados. Com isso, foi possível estruturar cada processo como uma sequência de nós interligados, nos quais cada etapa do ciclo é representada de maneira explícita. A biblioteca oferece componentes como `StateGraph`, `START` e `END`, que favorecem a organização lógica dos procedimentos e facilitam a implementação de decisões condicionais, fator relevante diante da necessidade de controle adaptativo (Jeong 2024);

d. **Docling**: empregado para converter manuais automotivos em PDF para o formato Markdown, etapa importante para viabilizar a extração de conteúdo textual estruturado e a indexação dos fragmentos. A escolha pelo Docling se deve à sua capacidade de preservar a organização hierárquica dos documentos e facilitar o processamento subsequente, especialmente em fluxos que demandam separação por seções, títulos e metadados (Auer et al. 2024);

[1] https://www.python.org/

e. **Pydantic**: empregado para validar e organizar dados utilizados na interação com modelos de linguagem, o que assegura que tais dados estejam no formato esperado para processamento (Ogli 2024);

f. **Typing**: oferece ferramentas para definição explícita de tipos no Python, o que contribui para tornar mais clara a estruturação do código e minimizar potenciais erros durante o desenvolvimento (Khan et al. 2021);

g. **Docker**: utilizado exclusivamente para gerenciar o contêiner do Milvus, situação na qual favorece a portabilidade e a reprodutibilidade no armazenamento e recuperação de *embeddings* em bases de dados vetoriais (Matthias & Kane 2015).

5.3 Operação

Nesta seção, detalham-se os procedimentos de preparação e execução do estudo, contemplando a configuração de parâmetros, a segmentação do texto, o uso da base Milvus e a geração de respostas às perguntas formuladas a partir dos manuais do Volkswagen Polo 2025 e do Fiat Argo 2023.

1) **Preparação**: para garantir consistência e comparabilidade entre as abordagens, definiram-se parâmetros gerais de segmentação e recuperação, tais como modelo de *embedding*, tamanho e sobreposição dos fragmentos, valor de k, métrica de similaridade e configuração do modelo gerador, conforme ilustra a Figura 5.1.

a. **Configuração de *Embeddings***: adotou-se o modelo `text-embedding-3-small`, da OpenAI, ajustado para gerar vetores a partir dos fragmentos obtidos na etapa de segmentação. Tal escolha preserva o equilíbrio entre custo computacional e qualidade vetorial, aspecto decisivo quando se processam grandes volumes de trechos extraídos dos documentos técnicos (OpenAI 2024*b*). Além disso, o modelo empregado apresenta custo até cinco vezes inferior ao de seu antecessor, o `text-embedding-ada-002` (OpenAI 2024*c*);

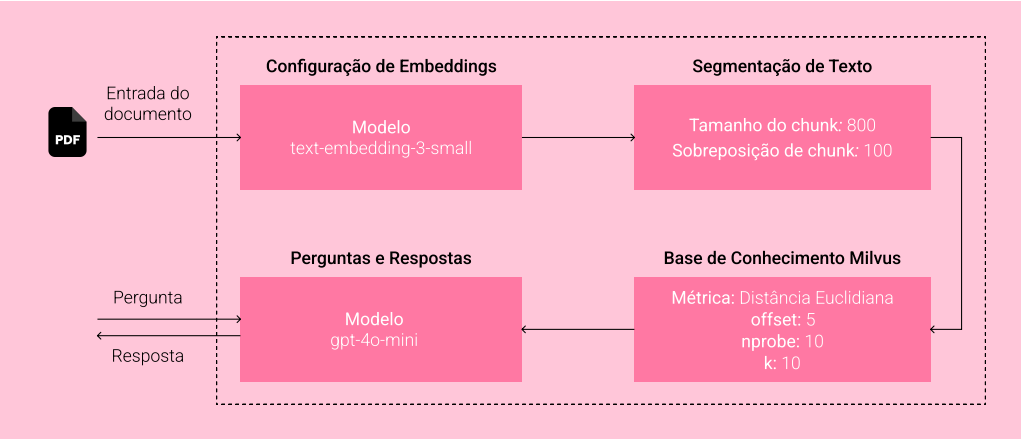
b. **Segmentação de Texto**: cada manual em PDF é convertido pelo Docling para Markdown, formato em texto simples que mantém a hierarquia de títulos. Em seguida, o conteúdo é particionado por esses títulos, preservando a estrutura original. Posteriormente, cada segmento é fracionado em blocos de até 800 caracteres, com sobreposição de 100 caracteres, configuração que se encontra na faixa recomendada para recuperar informações factuais em documentos técnicos (Gao et al. 2024). Essa sobreposição, equivalente a 12,5% do tamanho do bloco, segue a prática de 10–15% indicada para preservar a continuidade semântica entre fragmentos adjacentes (Microsoft 2024). Caso algum bloco ultrapasse 1000 caracteres, aplica-se nova subdivisão para manter o texto dentro do limite estabelecido;

c. **Base de Conhecimento Milvus**: em cada consulta, o sistema seleciona os dez fragmentos mais próximos do vetor da pergunta (`k = 10`) para formar o contexto de resposta. O mecanismo opera sobre um *índice vetorial*, isto é, uma estrutura

que armazena as representações (*embeddings*) de cada fragmento e permite localizar vizinhos semelhantes por meio de cálculos de distância. Nesse índice, define-se a Distância Euclidiana (L2) como métrica de similaridade. Na busca, utiliza-se `nprobe = 10`. Com esse valor, o Milvus examina dez partições internas do índice (*clusters*) antes de refinar as distâncias, mantendo o equilíbrio entre tempo de resposta e abrangência dos vizinhos recuperados;

d. **Perguntas e Respostas**: para a geração das respostas, foi utilizado o modelo `GPT-4o-mini`, reconhecido por sua capacidade avançada de compreensão e produção de texto.

Figura 5.1: Parâmetros de configuração do processo de preparação.



Fonte: Elaborada pela autora (2025).

2) **Execução**: elaboraram-se vinte perguntas, dez sobre o Volkswagen Polo 2025 e dez sobre o Fiat Argo 2023, de modo a cobrir situações de alerta no painel, manutenção preventiva, conectividade, conforto e uso de sistemas auxiliares. Cada questão foi submetida às seis variantes avaliadas — RAG Padrão, RAG com Gradiente Descendente, *Step-Back*, *Multi-Query*, *Self-RAG* Adaptativo e *Self-RAG* Adaptativo com Gradiente Descendente. As perguntas formuladas para cada manual são detalhadas a seguir.

Questões para o Manual do Volkswagen Polo 2025

1) Acendeu uma luz amarela no painel com um desenho de motor (parece um motorzinho). O que pode ser e o que eu devo fazer?

2) Como eu conecto meu celular Android no carro para usar os aplicativos na tela?

3) Qual é a calibragem certa dos pneus para o dia a dia e onde eu encontro essa informação no carro?

4) A luz do freio está acesa em vermelho e apitando! O que o manual diz sobre isso? É seguro continuar?

5) Meu carro tem aquele sistema que desliga o motor sozinho no semáforo para economizar combustível. Como ele funciona e tem alguma situação que eu não deva usar?

6) Como eu ajusto a hora no painel do carro?

7) Se furar o pneu, onde ficam o macaco e as ferramentas para trocar? E como eu faço a troca?

8) O carro tem diferentes modos de condução, tipo 'Eco' e 'Sport'. Qual a diferença entre eles e como eu seleciono?

9) Preciso puxar um reboque pequeno. Meu Polo pode fazer isso e tem alguma recomendação especial no manual?

10) Quando é a próxima revisão do carro? É por tempo ou por quilometragem?

Questões para o Manual do Fiat Argo 2023

1) Estou com um pneu furado na estrada! Onde fica o estepe, o macaco e as ferramentas, e quais os passos principais para trocar o pneu em segurança?

2) A bateria parece ter descarregado e o carro não liga. Como posso fazer a partida com uma bateria auxiliar ("chupeta") de outro veículo sem estragar nada?

3) Como eu sei qual a pressão correta para os pneus do meu carro e onde encontro essa informação?

4) A luz de temperatura do motor acendeu e está saindo vapor do capô! O que devo fazer?

5) Passei num buraco e meu carro parou de funcionar, não liga mais. Será que ativou alguma trava de segurança do motor? O que eu faço?

6) As luzes do ABS e do freio (vermelha com um ponto de exclamação) acenderam juntas no painel do Argo. É perigoso continuar dirigindo assim?

7) Se eu esquecer a chave dentro do carro, como o sistema reage ao tentar trancar as portas?

8) Meu carro às vezes mostra um aviso de 'HCSS' ou uma luzinha tipo uma 'mola' antes de dar a partida, especialmente quando está frio. Isso é normal? O que eu faço?

9) Quantos litros de combustível cabem no tanque do carro, incluindo a reserva?

10) Qual é a capacidade de óleo do motor para o Argo com motor 1.0 Flex, quando troco o filtro junto?

Para cada pergunta, compararam-se as respostas fornecidas pelas seis variantes, registrando-se pontuações de completude, fidelidade, relevância e aspectos de segurança.

5.4 Métricas de Avaliação

Nesta seção, são descritas as métricas adotadas para avaliar as respostas produzidas pela abordagem proposta, contemplando critérios como fidelidade ao contexto recuperado, relevância para a pergunta, completude das informações necessárias e segurança das respostas fornecidas. As métricas foram selecionadas e adaptadas com base na escala de Likert, frequentemente utilizada em avaliações qualitativas para capturar percepções detalhadas e sutis dos avaliadores (Likert 1932).

Para isso, foi realizada uma avaliação qualitativa assistida por *LLM-as-a-Judge*, na qual um modelo de linguagem atua como avaliador das respostas geradas. Assim, foi empregado o modelo `GPT-4o`, o qual foi escolhido com base em seu desempenho estabelecido em *benchmarks* para compreensão de linguagem natural e seguimento de instruções (OpenAI 2024*a*), de modo a assegurar um alto grau de confiabilidade na avaliação. Conforme Zheng et al. (2023), essa estratégia apresenta alta correlação com julgamentos humanos, com concordância superior a 80%. Essa alta taxa de concordância sugere que o *LLM-as-a-Judge* é um método escalável e explicável para aproximar as preferências humanas, as quais são muito caras e trabalhosas de obter por meios tradicionais. Além disso, aplica-se uma métrica automática de similaridade semântica (BERTScore) para comparar respostas de diferentes configurações do sistema RAG, conforme descrito a seguir.

1) **Fidelidade ao Contexto**: avalia o grau de correspondência entre as respostas geradas e o conteúdo dos manuais técnicos recuperados, buscando evitar inconsistências ou distorções. Em síntese, trata-se da aderência factual às informações das fontes utilizadas (Lewis et al. 2020). Tal definição encontra respaldo no conceito de *faithfulness* da literatura sobre RAG, que analisa se a resposta reflete de maneira adequada os documentos fornecidos (Shuster et al. 2021, Malin et al. 2024). Uma resposta com alta fidelidade (nota 5) apresenta todos os detalhes pertinentes de forma correta, sem introduzir elementos adicionais que não estejam embasados pelo contexto recuperado. Por outro lado, a fidelidade mínima (nota 1) é atribuída às respostas com informações imprecisas ou em desacordo com as fontes consultadas, especialmente no que se refere a especificações técnicas ou procedimentos críticos (Maynez et al. 2020). Diversos estudos ressaltam a relevância desse critério, uma vez que aplicações RAG surgiram com o intuito de reduzir a ocorrência de respostas incorretas e prevenir interpretações errôneas que possam implicar riscos (Lewis et al. 2020, Shuster et al. 2021).

2) **Relevância**: examina o quanto uma resposta atende diretamente à pergunta formulada pelo usuário (Korenčić et al. 2021). Em sistemas de perguntas e respostas técnicos especializados, é importante que a resposta corresponda exatamente ao problema apresentado. Assim, uma resposta totalmente relevante (nota 5) aborda integralmente o questionamento proposto, enquanto uma resposta irrelevante (nota 1) não responde adequadamente à pergunta, podendo até fornecer informações corretas, mas desvinculadas do tema específico questionado (Korenčić et al. 2021). Respostas desviantes, ainda que corretas, podem prejudicar a experiência de uso em sistemas técnicos especializados, justificando o rigor na avaliação deste critério.

3) **Completude**: avalia se as respostas contemplam todas as informações necessárias para solucionar de forma adequada a questão proposta (Li et al. 2024), distinguindo-se

em dois níveis: crítico e essencial. Informações críticas referem-se a avisos de segurança explícitos e pré-requisitos fundamentais, cuja omissão pode causar prejuízos ou riscos. Já informações essenciais englobam etapas e detalhes importantes, embora menos críticos que os primeiros. Essa divisão segue orientações de documentação técnica, que valorizam a exatidão e integralidade das instruções fornecidas (Banerjee & Lavie 2005). Uma resposta plenamente completa (nota 5) cobre integralmente ambos os níveis, enquanto uma resposta criticamente incompleta (nota 1) falha ao omitir informações críticas essenciais, comprometendo a segurança ou eficácia do procedimento.

3) **Segurança**: verifica se as respostas respeitam práticas automotivas seguras, evitando omissões de alertas essenciais e recomendações perigosas. Avaliam-se dois aspectos principais: primeiro, verifica-se a ausência de avisos críticos presentes nos manuais; segundo, identifica-se a inclusão de orientações que contrariem normas ou gerem riscos adicionais. Respostas totalmente seguras (nota 5) contemplam integralmente os alertas relevantes e não incluem práticas inseguras. Em contraste, respostas totalmente inseguras (nota 1) sugerem procedimentos perigosos ou ignoram avisos importantes. A relevância dessa métrica é corroborada por pesquisas sobre segurança técnica automotiva, destacando as consequências graves associadas a orientações inadequadas ou incompletas. Erros em informações técnicas podem afetar a confiança dos usuários e gerar riscos operacionais sérios (Winkelhake 2019).

4) **Similaridade Semântica (BERTScore)**: além das avaliações anteriores, emprega-se a métrica automática BERTScore para verificar a similaridade semântica entre respostas geradas por diferentes variantes de RAG. Em vez de se basear em correspondências literais ou superficiais com referências, o BERTScore utiliza *embeddings* contextuais de modelos BERT para avaliar a proximidade semântica entre pares de respostas. Valores elevados indicam forte convergência semântica, enquanto valores baixos sinalizam divergência (Zhang et al. 2019).

5) ***Score* Final**: o score final das respostas é calculado por meio de uma média ponderada das avaliações individuais (fidelidade, relevância, completude e segurança), utilizando pesos previamente estabelecidos com base na relevância relativa de cada critério em contextos automotivos. A escolha dos pesos (fidelidade: 40%, segurança: 30%, completude: 20%, relevância: 10%) reflete a prioridade dada à precisão factual e segurança operacional. Dessa forma, prioriza-se a precisão factual e segurança operacional devido às consequências que erros nessas áreas podem ocasionar.

Por fim, o material necessário para a reprodução do estudo encontra-se em repositório disponível no GitHub [2], que reúne as implementações das variantes de RAG, os arquivos de configuração, os notebooks de análise, as respostas produzidas por cada variante para as vinte perguntas, as métricas atribuídas pelo *LLM-as-a-judge*, além de outras informações úteis para replicação.

[2] https://github.com/conect2ai/rag-techniques-comparison

Seção 6

Resultados e Discussão

Nesta seção, apresentam-se os resultados da avaliação das diferentes variantes RAG. Inicialmente, a análise compara a qualidade e a consistência das respostas, destacando as variantes com desempenho superior em cada métrica. Em seguida, a discussão detalha os fatores associados a esse desempenho, com base nas quatro dimensões consideradas: fidelidade, relevância, completude e verificação de segurança. Para garantir uniformidade e imparcialidade na avaliação, essas métricas foram atribuídas automaticamente por um LLM, que seguiu critérios estruturados e padronizados para cada dimensão. Posteriormente, são examinados a robustez e o comportamento contextual das abordagens, avaliando o desempenho diante de diferentes tipos de pergunta. Por fim, a análise contempla o grau de similaridade entre as respostas produzidas pelas variantes, o que permite identificar convergências e divergências nos conteúdos gerados.

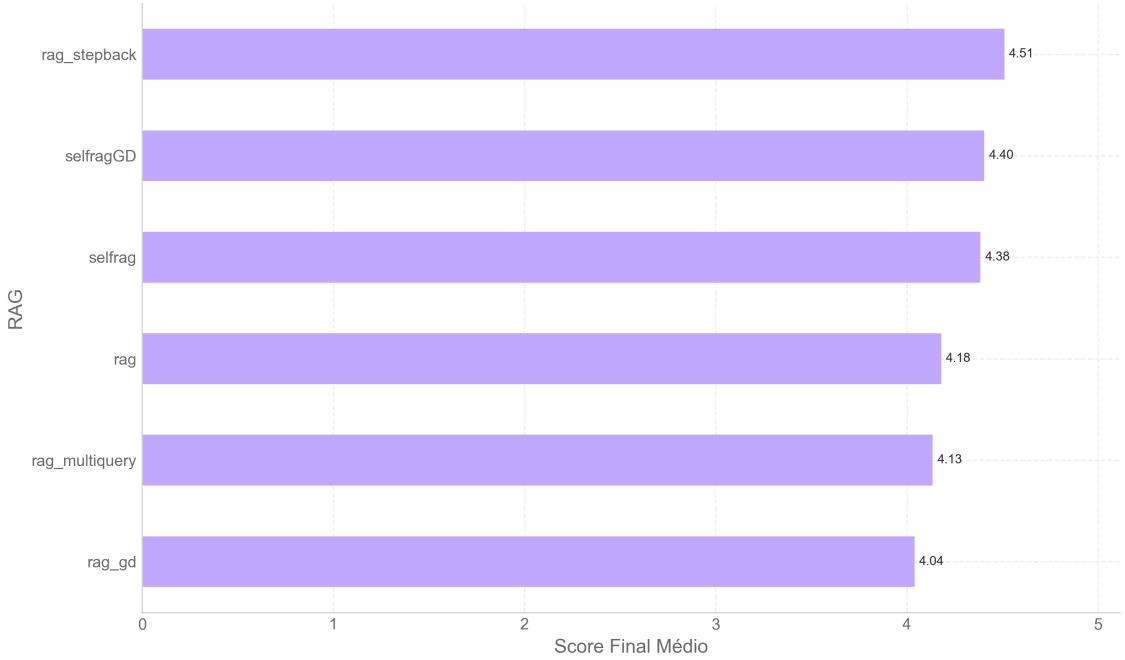
6.1 Qualidade e Consistência das Respostas

A avaliação inicial do desempenho foca em estabelecer uma comparação da qualidade e da consistência das respostas geradas por cada variante RAG. Para esse fim, considera-se como primeiro indicador a pontuação final média, apresentada na Figura 6.1. Com base nessa métrica, torna-se possível observar diferenças gerais entre as abordagens avaliadas e identificar tendências no comportamento de cada uma diante do conjunto de perguntas analisadas.

A partir dos dados, observa-se uma hierarquia de desempenho na qual a abordagem de *Step-Back Prompting* (rag_stepback) alcança a maior média (4.51), enquanto as seguida pelas variantes *Self-RAG* (selfragGD e selfrag) apresentam os valores subsequentes (4.40 e 4.38, respectivamente). Este resultado é um indicativo de desempenho superior para as arquiteturas que empregam técnicas mais elaboradas de recuperação ou geração. A abordagem *Step-Back*, por exemplo, utiliza um método que abstrai a pergunta original para uma consulta mais geral antes da busca. Tal estratégia tende a recuperar contextos com maior robustez e informação, o que pode ser um fator explicativo para sua alta pontuação média.

Além disso, a Figura 6.2 ilustra a distribuição de classificações (*rankings*) que cada sistema obteve. Este gráfico permite visualizar a frequência com que cada implementação foi a melhor (1ª posição), a segunda melhor (2ª posição), por exemplo.

Figura 6.1: *Score* Final Médio por RAG.

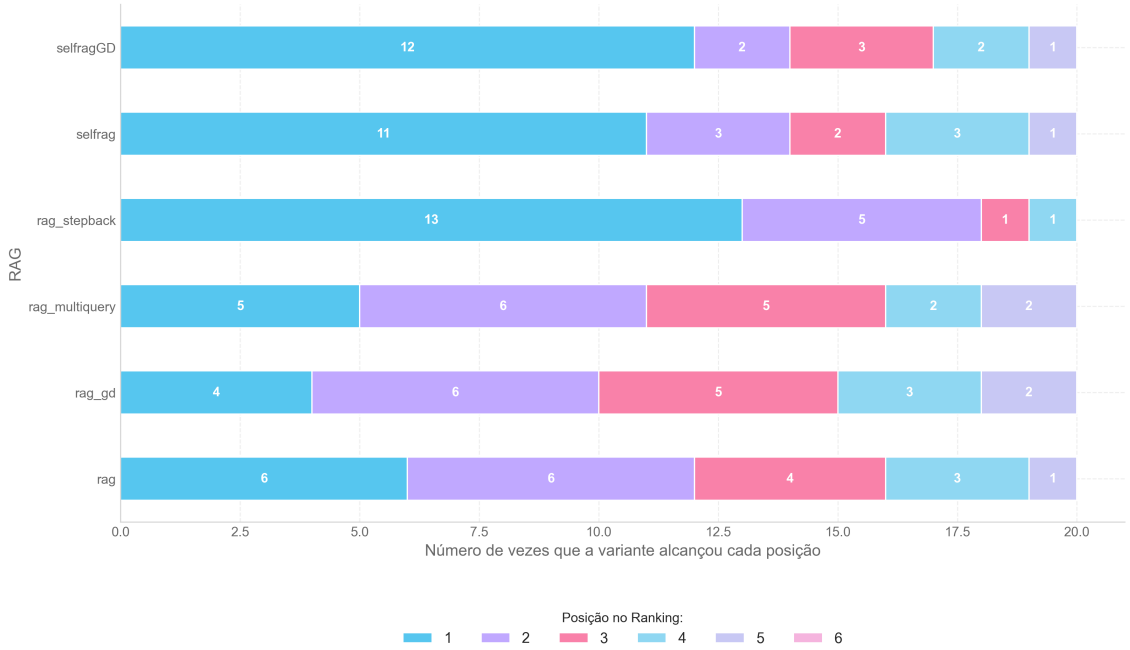


Fonte: Elaborada pela autora (2025).

Nesse sentido, a análise de *ranking* corrobora o desempenho superior do método *Step-Back*, que obteve a primeira posição na maioria das avaliações (13 de 20). Os sistemas baseados em *Self-RAG* também exibem um perfil de alta consistência. Dessa maneira, enquanto o sucesso da abordagem *Step-Back* parece residir na melhoria da recuperação de contexto, o da arquitetura *Self-RAG* pode ser atribuído à melhoria da geração. A técnica de auto-reflexão da *Self-RAG*, por exemplo, permite que o modelo avalie a si mesmo durante a escrita, o que pode promover maior fidelidade ao contexto recuperado. Ambas as estratégias, embora distintas, conduzem a uma performance de alta consistência. Em contraste, a heterogeneidade nos *rankings* do método *Multi-Query* (rag_multiquery) pode ser explicada por sua abordagem: a geração de múltiplas perguntas pode, em alguns casos, melhorar a busca, mas em outros, pode introduzir ruído ao buscar por tópicos tangenciais, tornando sua performance menos estável.

A Figura 6.3 sintetiza a relação entre a performance (eixo x) e a consistência (eixo y, onde valores mais baixos indicam maior estabilidade), o que permite uma avaliação integrada do desempenho. As arquiteturas *Step-Back* e *Self-RAG* (identificadas nos gráficos como rag_stepback, selfragGD e selfrag) agrupam-se no quadrante inferior direito, cenário que representa alta performance e alta consistência. Por um lado, esta visualização apoia a ideia de que as otimizações aplicadas por estas abordagens, seja na recuperação, seja na geração de texto, elevam a qualidade média das respostas e, de forma conjunta, reduzem a variabilidade do desempenho. Por outro lado, a alta inconsistência de sistemas como o *Multi-Query* e a abordagem RAG com Gradiente Descendente (rag_gd) pode ser

Figura 6.2: Consistência de Desempenho – Frequência de Posições no *Ranking*.



Fonte: Elaborada pela autora (2025).

atribuída à sua maior dependência de uma etapa de recuperação única e mais simples, que é mais suscetível a falhas. Assim, observa-se que métodos que aplicam estratégias de otimização para contornar limitações na recuperação ou na geração obtiveram desempenho mais elevado e mantiveram maior estabilidade ao longo das perguntas.

Ademais, é possível perceber que a variante RAG convencional ocupa uma posição próxima à linha divisória entre estabilidade e instabilidade. Em razão disso, sugere-se que a abordagem não apresenta oscilações acentuadas, mas também não se destaca em nenhuma das dimensões avaliadas quando comparada às variantes que adotam estratégias adicionais nas etapas de recuperação ou geração.

6.2 Comparação das Quatro Métricas de Avaliação

Após a análise da qualidade e consistência das respostas, investiga-se os fatores que contribuem para o desempenho observado entre as variantes avaliadas. Desse modo, sobre dimensões que influenciam a utilidade das respostas em consultas técnicas, isto é, a capacidade de cobrir de maneira adequada os elementos informacionais esperados, o grau de fidelidade em relação ao conteúdo de referência e a pertinência das informações apresentadas. As métricas de completude, verificação de segurança, fidelidade e relevância servem, nesse contexto, como indicadores para examinar essas dimensões em maior profundidade.

A Figura 6.4 apresenta a média das pontuações obtidas por cada variante RAG nas

Figura 6.3: Análise de Quadrantes – Performance vs. Consistência por RAG.

Fonte: Elaborada pela autora (2025).

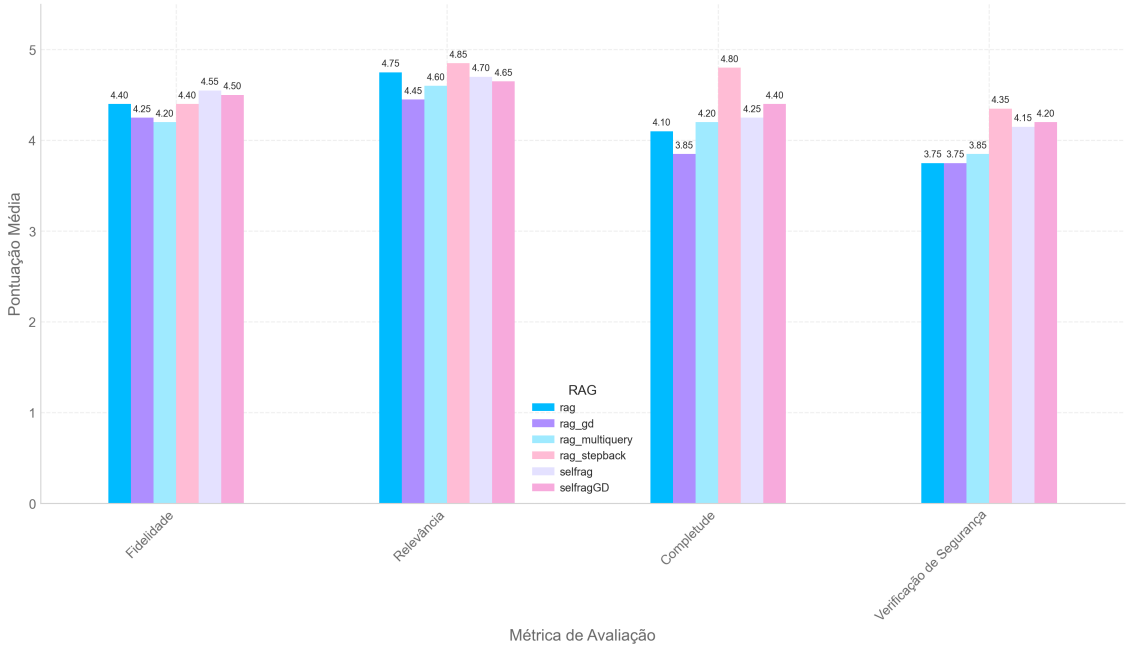
quatro métricas consideradas. Observa-se que, de modo geral, o método *Step-Back* e as variantes *Self-RAG* apresentam as maiores médias em completude, verificação de segurança e fidelidade. Em completude, *Step-Back* se destaca com a maior média (4,80), seguido pelas abordagens *Self-RAG*. Na verificação de segurança, esse grupo adaptativo também supera as demais variantes, com valores superiores a 4, enquanto os métodos convencionais permanecem abaixo desse patamar.

No que se refere à completude, o método *Step-Back* apresenta a média mais elevada (4,80), seguido pelas variantes *Self-RAG*, que alcançam 4,40 e 4,25. Por outro lado, os métodos convencionais apresentam valores mais baixos, o que indica maior frequência de omissões de informações consideradas essenciais para o atendimento pleno das perguntas. Nesse sentido, essa diferença pode estar relacionada à ausência de mecanismos de revisão ou decomposição, presentes nas variantes adaptativas.

Em relação à verificação de segurança, observa-se que a abordagem *Step-Back* apresenta a média mais elevada (4,35), sendo seguida por *Self-RAG* com Gradiente Descendente (4,20) e *Self-RAG* (4,15). Além disso, a variante *Multi-Query* registra média intermediária (3,85). Por outro lado, os métodos convencionais RAG e RAG com Gradiente Descendente apresentam os menores resultados nessa métrica, ambos com 3,75, o que sugere maior frequência de omissões de informações relevantes para segurança.

Quanto à fidelidade, as diferenças entre as variantes são mais discretas. Nota-se que

Figura 6.4: Pontuação Média das Métricas LLM por RAG.



Fonte: Elaborada pela autora (2025).

Self-RAG e *Self-RAG* com Gradiente Descendente alcançam as maiores médias (4,55 e 4,50, respectivamente). Em seguida, RAG convencional e *Step-Back* registram valores iguais (4,40), enquanto *Multi-Query* apresenta o menor resultado (4,20). Dessa forma, percebe-se uma leve vantagem das abordagens adaptativas nessa métrica, embora a proximidade dos valores indique que todas as variantes mantêm correspondência relativamente alta com o conteúdo de referência.

De modo geral, os resultados indicam que variantes que adotam estratégias de reescrita, decomposição ou autoavaliação apresentam desempenho superior em completude, verificação de segurança e, em menor grau, fidelidade. Embora as diferenças em fidelidade sejam menos expressivas, observa-se que as médias permanecem elevadas em todas as variantes, refletindo aderência consistente ao conteúdo de referência. Por outro lado, as abordagens convencionais apresentam limitações mais evidentes, especialmente no que se refere à cobertura dos elementos essenciais e à atenção a aspectos de segurança, o que reforça a importância de mecanismos adicionais para aprimorar a qualidade das respostas geradas.

6.3 Análise de Robustez e Desempenho Contextual

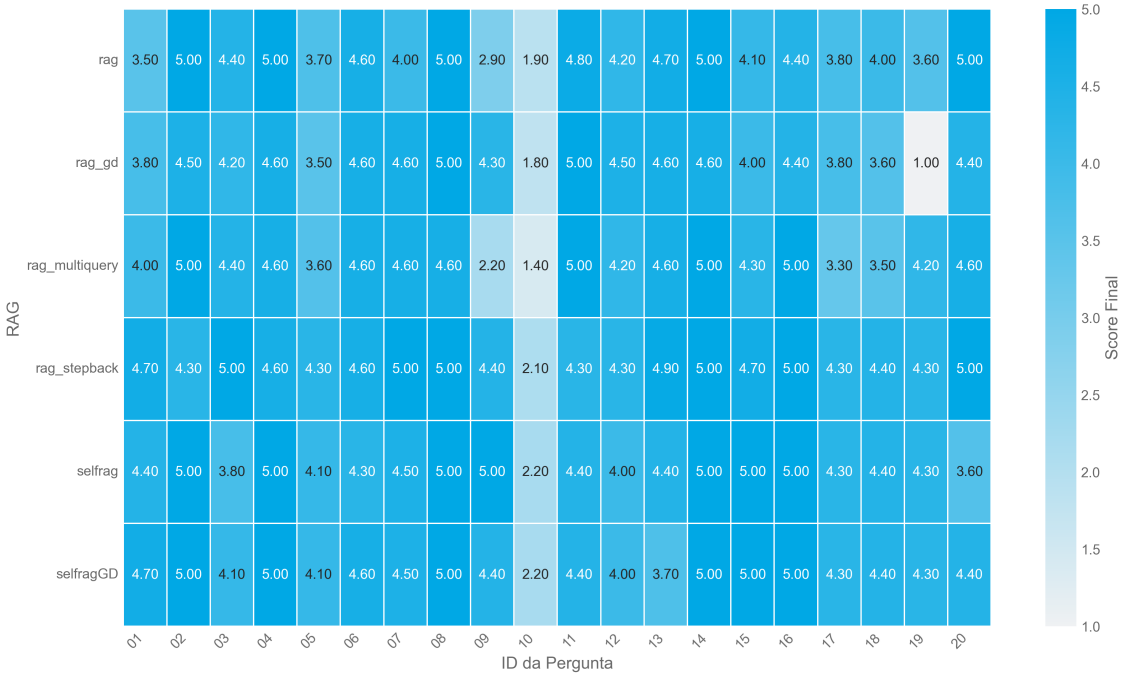
Para além das médias gerais, é importante examinar como cada abordagem se comporta diante de diferentes tipos de perguntas. A Figura 6.5 apresenta um panorama das pontuações finais obtidas por cada variante RAG ao longo das vinte perguntas avaliadas.

A partir disso, é possível observar a distribuição dos desempenhos em cenários variados, o que permite identificar padrões de estabilidade, eventuais quedas de desempenho e sensibilidade a perguntas específicas.

A Figura 6.5 ilustra a distribuição das pontuações finais obtidas por cada abordagem ao longo das vinte perguntas avaliadas. Observando o comportamento das variantes diante desse espectro de cenários, percebe-se que métodos adaptativos, como *Step-Back* e *Self-RAG*, mantêm desempenho elevado e estável na maior parte das situações, independentemente do grau de complexidade ou do contexto da pergunta. Por outro lado, as abordagens convencionais, como RAG e RAG com Gradiente Descendente, apresentam maior oscilação nos resultados. Observa-se, em especial, uma queda mais acentuada em determinadas perguntas, sobretudo naquelas que exigem múltiplos passos, interpretação de situações de emergência ou menção explícita a procedimentos de segurança.

Adicionalmente, algumas perguntas apresentam-se como pontos de maior dificuldade para todas as variantes, com queda nas avaliações atribuídas, o que pode indicar limitações das abordagens avaliadas, além de possíveis ambiguidades ou lacunas nas informações presentes nos manuais. Em síntese, a análise detalhada desse comportamento evidencia situações em que as estratégias adotadas mantêm certa estabilidade frente à diversidade de perguntas. Por outro lado, revela cenários nos quais ajustes metodológicos ou revisão das fontes podem ser necessários para aprimorar o desempenho das variantes.

Figura 6.5: Heatmap de Performance – Score Final por RAG e Pergunta.

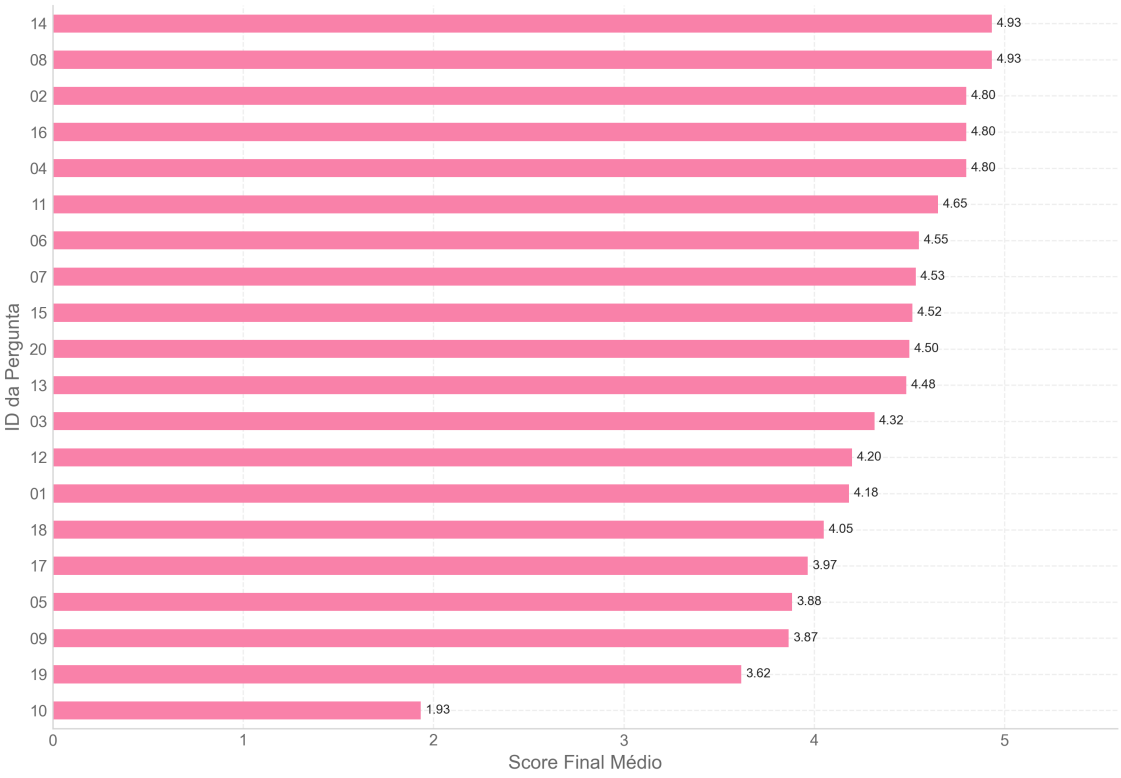


Fonte: Elaborada pela autora (2025).

Além disso, a Figura 6.6 evidencia que a pergunta 10, relacionada ao intervalo da próxima revisão e pertencente ao conjunto do Polo (perguntas 1 a 10), apresenta o menor

score médio (1,93). Em seguida, surgem a pergunta 19 do Argo, sobre a capacidade total do tanque de combustível (3,62), e a pergunta 9 do Polo, que trata de requisitos para reboque (3,87). Esses três casos envolvem consultas sobre revisões periódicas, capacidades numéricas ou limites mecânicos, tópicos que, com frequência, aparecem de forma resumida ou dispersa em manuais automotivos. A possível necessidade de localizar valores específicos em tabelas ou seções técnicas pode ter contribuído para a queda de desempenho observada, ainda que outras variáveis (como ambiguidade da pergunta ou ausência explícita da informação) também possam exercer influência.

Figura 6.6: Score Final Médio por ID da Pergunta.



Fonte: Elaborada pela autora (2025).

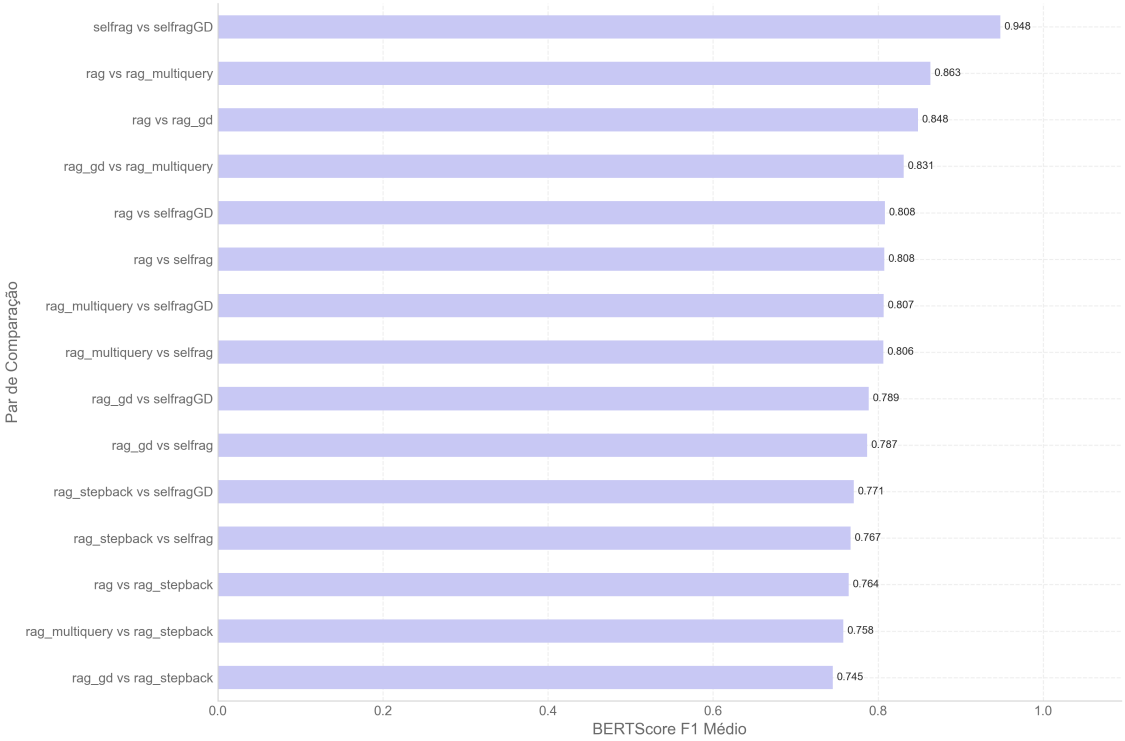
Em contraste, perguntas como a 14 do Argo (procedimento diante de superaquecimento), a 8 do Polo (diferentes modos de condução) e a 2 do Argo (partida auxiliar com bateria) obtêm escores médios próximos ao valor máximo. Possivelmente, esses temas contam com orientações mais objetivas ou procedimentos padronizados nos manuais, o que tende a simplificar a recuperação da informação pelas variantes. Desse modo, o desempenho contextual mostra-se mais estável quando a consulta refere-se a instruções claras e concentradas em um único trecho do documento. Em contraste, questões que exigem síntese de dados dispersos ou menos padronizados continuam a representar um desafio comum entre as abordagens avaliadas.

6.4 Similaridade entre as Respostas Geradas

Para além da análise de desempenho individual das variantes, examinar o grau de similaridade entre as respostas produzidas por diferentes métodos permite avaliar a proximidade semântica entre os conteúdos gerados. Nesse contexto, a comparação pode indicar a presença de respostas redundantes ou, em sentido oposto, revelar variações relevantes decorrentes das estratégias adotadas por cada abordagem. A Figura 6.7 apresenta os valores médios de BERTScore F1 calculados para cada par de variantes, fornecendo uma medida objetiva do quanto as respostas assemelham-se do ponto de vista semântico.

No conjunto dos pares analisados, *Self-RAG* e *Self-RAG* com Gradiente Descendente apresentam o índice mais elevado de similaridade (0,948), o que indica produção de respostas praticamente idênticas em conteúdo e forma. De modo semelhante, RAG convencional em combinação com *Multi-Query* (0,863) ou com RAG com Gradiente Descendente (0,848) exibe valores altos, sugerindo convergência semântica entre essas abordagens tradicionais.

Figura 6.7: Similaridade das respostas entre pares de variantes.



Fonte: Elaborada pela autora (2025).

Por outro lado, as combinações que incluem *Step-Back* apresentam índices de similaridade mais moderados. O par *Step-Back* e *Self-RAG* com Gradiente Descendente alcança 0,771, enquanto *Step-Back* e RAG com Gradiente Descendente registra o menor valor observado (0,745). Tal diferença pode estar relacionada ao mecanismo de reescrita de perguntas empregado no **Step-Back**, que tende a introduzir variações lexicais e estruturais nas respostas.

6.5 Discussão

Nesta seção, os resultados obtidos são interpretados e discutidos conforme as três Questões de Pesquisa (QP) que nortearam este trabalho.

Em relação à QP 1, que investiga a influência de etapas adicionais de processamento na correspondência semântica, os resultados indicam que abordagens como *Step-Back* e *Self-RAG* alcançam níveis superiores de fidelidade e completude. O desempenho superior dessas variantes sugere que a abstração da consulta (*Step-Back*) ou a autorreflexão durante a geração (*Self-RAG*) facilitam a seleção de evidências mais pertinentes do contexto. Isso, por sua vez, permite a incorporação de detalhes críticos na resposta final, elevando sua correspondência com o conteúdo de referência e superando as variantes convencionais.

No que tange à QP 2, sobre o impacto da Engenharia de *Prompts* na qualidade textual, os resultados indicam que estratégias de *Engenharia de Prompts* influenciam diretamente a forma como os contextos são selecionados. No caso do método *Step-Back*, a reformulação da pergunta original pode ter contribuído para direcionar a recuperação a trechos mais informativos, o que se refletiu em pontuações mais altas nas métricas de completude e verificação de segurança, em comparação às demais variantes. Por outro lado, a abordagem *Multi-Query* apresentou variações de desempenho ao longo das perguntas, indicando que a formulação de múltiplas consultas nem sempre resulta em maior qualidade e, em alguns casos, pode ampliar o escopo da busca de maneira pouco controlada.

Quanto à QP 3, que questiona a capacidade das técnicas de superar limitações das abordagens convencionais, os dados sugerem que a adoção de estratégias como decomposição da consulta, reescrita e autoavaliação contribuiu para reduzir limitações observadas nas variantes convencionais. Embora ainda tenham ocorrido quedas de desempenho em determinadas perguntas, as abordagens adaptativas demonstraram maior estabilidade e médias superiores nas quatro métricas avaliadas. Dessa forma, os resultados obtidos indicam que métodos que combinam mecanismos de controle na geração e recuperação tendem a oferecer respostas mais consistentes quando aplicados a cenários de consulta técnica, com ênfase no contexto automotivo.

Por fim, a análise de similaridade semântica (BERTScore) complementa a discussão ao revelar diferentes níveis de convergência. As altas pontuações entre variantes com arquiteturas parecidas (como as duas de *Self-RAG*) sugerem processos de geração muito similares. Em oposição, a baixa similaridade das respostas do *Step-Back* com as demais reforça como sua estratégia de reescrita da consulta produz um resultado textualmente distinto. De forma geral, os resultados consolidam a evidência de que o uso de estratégias adicionais, seja na recuperação, seja na geração, está diretamente vinculado a ganhos de desempenho e estabilidade em sistemas RAG aplicados a manuais automotivos.

Seção 7

Conclusão

Nesta seção, sintetizam-se as principais evidências obtidas a partir da comparação de seis variantes de *Retrieval-Augmented Generation* em consultas a manuais automotivos.

O estudo desenvolvido concentrou-se na verificação de três questões de pesquisa: a influência de etapas adicionais de processamento na correspondência semântica das respostas; o impacto da Engenharia de *Prompts* na qualidade textual; e a capacidade dessas técnicas de superar limitações de abordagens convencionais no domínio automotivo.

Para atender a essas questões, foi conduzido um experimento que comparou as variantes RAG Convencional, RAG com Gradiente Descendente, *Multi-Query*, *Step-Back*, *Self-RAG* e *Self-RAG* com Gradiente Descendente. No processo de avaliação, foram empregadas métricas para analisar a fidelidade ao contexto, a relevância em relação à pergunta, a completude das informações e a segurança das respostas, com atribuições realizadas automaticamente por um modelo de linguagem atuando como avaliador. Além disso, calculou-se o BERTScore para medir a proximidade semântica entre as respostas. O experimento incluiu vinte perguntas, sendo dez baseadas no manual do Volkswagen Polo 2025 e dez no manual do Fiat Argo 2023.

Os resultados indicaram que a adoção de etapas adicionais de processamento, como reformulação e autoavaliação, favorece maior aderência ao conteúdo de referência (QP1). Estratégias de Engenharia de *Prompts*, por sua vez, influenciaram a forma como os contextos foram selecionados, afetando diretamente a completude e a segurança das respostas (QP2). Além disso, variantes com mecanismos adaptativos apresentaram desempenho mais estável diante de diferentes tipos de pergunta, mitigando limitações identificadas nas abordagens convencionais (QP3). Por fim, a análise de similaridade semântica contribuiu para distinguir os comportamentos das variantes, com maior convergência entre aquelas que compartilham estruturas internas semelhantes, como *Self-RAG* e sua versão com Gradiente Descendente, e menor proximidade entre métodos que empregam reformulação mais profunda, como o *Step-Back*.

Em síntese, os resultados obtidos indicam que abordagens de *Retrieval-Augmented Generation* que incorporam mecanismos adicionais de reformulação, decomposição ou autoavaliação tendem a produzir respostas mais completas, seguras e aderentes ao contexto consultado. Além disso, as análises de similaridade evidenciaram que essas estratégias afetam a proximidade semântica entre as respostas geradas. Assim, esses resultados refutam a hipótese nula, pois evidenciam ganhos na qualidade das respostas geradas quando se aplicam variações na arquitetura do RAG e técnicas de Engenharia de *Prompts*.

Dessa forma, a hipótese alternativa fica sustentada, indicando que tais ajustes elevam o alinhamento semântico, a completude das informações e a consistência das respostas em cenários de consulta técnica.

Diante desses achados, estudos futuros podem, entre outras possibilidades, explorar três linhas. Em primeiro lugar, pode-se desenvolver pipelines híbridos nos quais a recuperação ocorre no próprio dispositivo e a geração textual é delegada a um serviço em nuvem, combinação que reduz a latência e diminui a carga de processamento local. Em segundo lugar, pode-se examinar o desempenho de técnicas baseadas em raciocínio explícito, verificando se cadeias de inferência elevam a qualidade das respostas em consultas técnicas e esclarecem o processo de obtenção de informação. Por fim, pode-se criar um módulo de seleção que determine, em cada consulta, se *Step-Back* ou *Self-RAG* atende melhor às características da pergunta, considerando extensão, especificidade e terminologia empregada.

7.1 Artigos Aprovados

O desenvolvimento deste trabalho originou duas publicações, como apresentadas a seguir:

Artigo 1: Medeiros, T.; Medeiros, M.; Azevedo, M.; Silva, M.; Silva, I.; e Costa, D. G. **Analysis of language-model-powered Chatbots for query resolution in PDF-based automotive manuals**. *Vehicles*, vol. 5, no. 4, pp. 1384–1399, 2023.

Neste estudo, foi analisada a aplicação de Modelos de Linguagem de Grande Escala em *chatbots* para a resolução de consultas em manuais automotivos no formato PDF. A pesquisa realizou uma avaliação comparativa de três abordagens diferentes de interação com os manuais, considerando métricas como precisão das respostas, eficiência de custos e experiência do usuário. Os resultados mostram que os *chatbots* analisados podem simplificar o acesso a informações técnicas complexas e contribuir para melhorar a manutenção e o suporte técnico de veículos. Contudo, os desafios de interpretar elementos visuais e otimizar custos foram identificados como pontos que necessitam de maior refinamento. O artigo também destaca as implicações práticas e os benefícios potenciais dessa tecnologia no setor automotivo.

Artigo 2: Medeiros, T.; Medeiros, M.; Machado, H.; Alves, G.; Silva, M.; e Silva, I. **Integração de Modelos de Linguagem de Grande Escala com a técnica RAG para a Simplificação de Consultas a Manuais Automotivos**. Anais da Sociedade Brasileira de Automática, 2024.

Neste artigo, foi proposta uma abordagem para integrar LLMs com a técnica de RAG, visando simplificar consultas a manuais automotivos. Para isso, a proposta combina a capacidade de geração de texto dos LLMs com a recuperação precisa de informações por meio do RAG, com o intuito de melhorar a eficiência e a clareza no acesso a dados técnicos complexos. Como resultado, os dados obtidos indicam que a técnica proposta pode reduzir a complexidade das consultas, facilitando, dessa forma, o entendimento e a utilização de manuais automotivos por profissionais e usuários finais. Por fim, o trabalho

discute as implicações práticas da aplicação dessa integração em cenários reais, sugerindo seu potencial para otimizar processos de manutenção e suporte técnico na indústria automotiva.

Além disso, há colaborações em outras áreas, conforme evidenciado nos artigos listados a seguir.

Artigo 1: Azevedo, M.; Andrade, M.; Medeiros, M.; Medeiros, T.; Silva, M.; e Silva, I. **Optimizing Vehicle IoT Systems: SUMO-Digital Twin Performance Analysis**. IEEE MetroInd4.0&IoT (MetroInd), 2024.

Neste trabalho, propõe-se o uso de uma arquitetura de dados veiculares avaliada por meio do simulador de tráfego urbano SUMO, com o objetivo de analisar sua eficiência e confiabilidade em aspectos como utilização de recursos do sistema, taxa de transferência, latência e integridade dos dados. Para isso, a abordagem baseia-se em uma arquitetura composta por um servidor configurado com Node-RED para coleta e pré-processamento de dados, uma base de dados Timescale para armazenamento de séries temporais e uma API para análise de dados. Nesse contexto, foram realizadas simulações iterativas com 91.296 veículos simultâneos, o que resultou no total de 273.888 veículos ao longo de três rodadas de experimentos. Com relação aos resultados preliminares, observa-se um aumento significativo no uso da CPU, com picos de até 99,9% de utilização, além de uma taxa média de transmissão de 297 dados por segundo, alcançando até 543 dados por segundo em determinados momentos. Apesar de os resultados sugerirem que a arquitetura é capaz de lidar com grandes volumes de dados em tempo real, é importante destacar que análises adicionais são necessárias para confirmar sua robustez e confiabilidade em cenários mais amplos e complexos.

Artigo 2: Andrade, M.; Medeiros, M.; Medeiros, T.; Silva, M.; e Silva, I. **Inferring Driver Behavior Profiles Using Digital Twins in Simulated Environments**. American Workshop on Safe and Secure Vehicles, 2024.

Neste estudo, foi investigado o uso de *digital twins* em ambientes simulados para compreender perfis de comportamento de motoristas, com foco em segurança viária e eficiência de combustível. Para isso, o Euro Truck Simulator 2 (ETS2) foi utilizado para coletar dados de telemetria, como velocidade, aceleração, RPM e consumo de combustível. Esses dados, por sua vez, foram integrados a um servidor de telemetria baseado em APIs REST e, em seguida, analisados com Python, o que permitiu a criação de métricas, como a área do radar, utilizadas para identificar padrões de direção. Nesse contexto, dois cenários foram simulados: um estilo de direção cauteloso, que priorizou a estabilidade, e outro agressivo, caracterizado por acelerações bruscas. Como resultado, verificou-se que o estilo cauteloso apresentou menor variação na métrica de radar e maior segurança. Em contrapartida, o estilo agressivo revelou picos significativos na métrica de radar, ocasionando maior desgaste do veículo. Assim, o estudo conclui que simuladores são ferramentas eficazes para a análise de comportamento de motoristas e, além disso, sugere o uso de algoritmos de aprendizado de máquina para prever comportamentos em tempo real e ampliar a aplicação em diferentes contextos e tipos de veículos.

Artigo 3: Medeiros, M.; Andrade, M.; Medeiros, T.; Silva, M.; e Silva, I. **Anomaly**

Detection in Simulated Vehicle Dynamics Using BeamNG.tech and the TEDA Framework. *American Workshop on Safe and Secure Vehicles*, 2024.

Neste trabalho, apresenta-se uma abordagem que integra a plataforma de simulação BeamNG.tech, um sistema de coleta de dados veiculares e o *framework Typicality and Eccentricity Data Analytics* (TEDA) para a detecção de anomalias em dados de velocidade de veículos. Primeiramente, a metodologia envolve a criação de um ambiente de simulação personalizado. Em seguida, aplica-se o TEDA para análise de dados em tempo real, além de garantir a transmissão eficiente dos dados para processamento e armazenamento, utilizando Node-RED e TimescaleDB. Para avaliar a abordagem, foi realizado um estudo de caso em que um motorista percorreu uma rota predefinida, com variações intencionais de velocidade, simulando comportamentos atípicos. Como resultado, o TEDA identificou 1.110 anomalias em 6.388 amostras de velocidade, correspondendo a 17,38% dos dados, o que evidencia a eficácia na detecção de padrões incomuns. Por fim, a integração proposta viabiliza análises em tempo real, com potencial para contribuir com o desenvolvimento de sistemas de assistência ao motorista e tecnologias de segurança automotiva.

Artigo 4: Andrade, M., Medeiros, T., Silva, M., e Silva, I. **Aplicação de Modelos de Linguagem de Grande Escala e Agentes Inteligentes com SUMO para Automatizaçao de Simulações de Tráfego Urbano**. Anais da Sociedade Brasileira de Automática, 2024.

Neste trabalho, investigou-se a integração de Modelos de Linguagem de Grande Escala com o SUMO, com o objetivo de explorar o potencial de automatização do processo de simulação urbana. Para isso, a abordagem proposta busca facilitar a configuração de cenários, a análise de resultados e a proposição de soluções adaptativas. Além disso, um estudo de caso foi conduzido com dados da cidade de Natal-RN, Brasil, cujos resultados sugerem a viabilidade dos LLMs para interpretar dados e automatizar processos. Por fim, os resultados indicaram que a metodologia foi capaz de configurar cenários de tráfego de maneira autônoma, realizar análises em tempo real e propor soluções adaptativas baseadas nos padrões detectados.

Artigo 5: Andrade, M.; Medeiros, M.; Medeiros, T.; Azevedo, M.; Silva, M.; Costa, Daniel G.; e Silva, I. **On the Use of Biofuels for Cleaner Cities: Assessing Vehicular Pollution through Digital Twins and Machine Learning Algorithms**. *Sustainability*, v. 16, p. 708, 2024.

Neste trabalho, propõe-se uma metodologia que utiliza algoritmos de aprendizado de máquina e gêmeos digitais para estimar e comparar as emissões de CO_2 de veículos movidos a etanol e gasolina. O objetivo é fornecer dados que apoiem a transição para uma matriz energética mais sustentável no setor de transportes. Para isso, a abordagem integra a coleta de dados do mundo real por meio de diagnósticos a bordo (OBD-II) e smartphones, utilizando um algoritmo de aprendizado não supervisionado para remover anomalias nos dados. Adicionalmente, um gêmeo digital do cenário de tráfego é criado com o simulador SUMO, e modelos de aprendizado de máquina supervisionado são treinados para permitir análises de poluição realistas no ambiente simulado. Um estudo de caso foi conduzido em um percurso de aproximadamente 13 km na cidade de Natal, Brasil, com um

veículo flex. Os resultados mostraram que o etanol emite menos CO_2 do que a gasolina. Por fim, a metodologia demonstrou a viabilidade do uso de gêmeos digitais para análises de poluição, uma vez que os resultados da simulação apresentaram notável conformidade com os dados coletados nos cenários reais.

Artigo 6: Flores, T. K. S.; Silva, I.; Azevedo, M. B.; Medeiros, T. A.; Medeiros, M. A.; Costa, D. G.; Ferrari, P.; e Sisinni, E. **Advancing Tiny Machine Learning Operations: Robust Model Updates in the Internet of Intelligent Vehicles**. IEEE Micro, v. 45, n. 1, p. 76-86, 2025.

Neste trabalho, é proposta uma metodologia de operações de aprendizado de máquina para dispositivos de baixíssimo consumo (TinyMLOps) no contexto da Internet de Veículos Inteligentes, utilizando microcontroladores ESP32. O objetivo é estabelecer um fluxo de trabalho para a atualização robusta de modelos de aprendizado de máquina em scanners de diagnóstico veicular (OBD-II) para facilitar melhorias contínuas e lidar com o problema de desvio de conceito (*concept drift*). Para isso, a abordagem utiliza uma estação de rádio (RS) fixa, que armazena a versão mais recente do modelo, e um dispositivo OBD-II no veículo. A RS se comunica com o OBD-II via protocolo Wi-Fi ESP-NOW, verifica se o modelo está desatualizado e, em caso afirmativo, transmite a nova versão de forma sequencial e segura. A metodologia foi validada com um experimento de atualização de um modelo de predição de emissão de CO_2, usando dados coletados de quatro veículos em um percurso de 5 km na cidade de Natal, Brasil. Os resultados demonstraram uma melhoria no desempenho do modelo após a atualização, com o erro médio absoluto e o desvio padrão caindo de 0.49 e 1.1 para 0.012 e 0.082, respectivamente. Por fim, o estudo evidencia a eficácia do sistema em termos de latência, consumo de energia e robustez na transmissão, validando a abordagem para aplicações do mundo real.

Referências Bibliográficas

Adamopoulou, Eleni & Lefteris Moussiades (2020), An overview of chatbot technology, *em* 'IFIP international conference on artificial intelligence applications and innovations', Springer, pp. 373–383.

Amato, Flora, Mattia Fonisto, Marco Giacalone & Carlo Sansone (2023), 'An Intelligent Conversational Agent for the Legal Domain', *Information (Switzerland)* **14**(6).

Aribas, Erke & Evren Daglarli (2024), 'Transforming Personalized Travel Recommendations: Integrating Generative AI with Personality Models', *Electronics* **13**(23), 4751.

Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil & Hannaneh Hajishirzi (2023), 'Self-rag: Learning to retrieve, generate, and critique through self-reflection', *arXiv preprint arXiv:2310.11511* .

Atombo, Charles, Maxwell Selase Akple, Arnold Dumenya & Paul Adzimahe (2025), 'Comprehensibility and impact of vehicle dashboard indicator light symbols on drivers' preventive maintenance compliance', *PloS one* **20**(5), e0323386.

Auer, Christoph, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla et al. (2024), 'Docling technical report', *arXiv preprint arXiv:2408.09869* .

Banerjee, Satanjeev & Alon Lavie (2005), METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, *em* 'Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization', pp. 65–72.

Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill et al. (2021), 'On the opportunities and risks of foundation models', *arXiv preprint arXiv:2108.07258* .

Bourne, Keith (2024), *Unlocking Data with Generative AI and RAG*, Packt Publishing, Birmingham, UK. First edition, published in September 2024.

Boykis, Vicki (2023), *What are Embeddings*, Colophon. Disponível em: https://vickiboykis.com/what_are_embeddings/. Acesso em: 20 de dezembro de 2024.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al. (2020), 'Language models are few-shot learners', *Advances in neural information processing systems* **33**, 1877–1901.

Cai, Jing, Alex E Hadjinicolaou, Angelique C Paulk, Daniel J Soper, Tian Xia, Alexander F Wang, John D Rolston, R Mark Richardson, Ziv M Williams & Sydney S Cash (2025), 'Natural language processing models reveal neural dynamics of human conversation', *Nature Communications* **16**(1), 3376.

Cardenas, Erika & Leonie Monigatti (2024), 'What is Agentic Retrieval-Augmented Generation (RAG)?', Disponível em: https://weaviate.io/blog/what-is-agentic-rag. Acesso em: 8 de janeiro de 2025.

Chandrasekhar, Achuth, Jonathan Chan, Francis Ogoke, Olabode Ajenifujah & Amir Barati Farimani (2024), 'AMGPT: A large language model for contextual querying in additive manufacturing', *Additive Manufacturing Letters* **11**, 100232.

Chen, Danqi, Adam Fisch, Jason Weston & Antoine Bordes (2017), 'Reading wikipedia to answer open-domain questions', *arXiv preprint arXiv:1704.00051* .

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova (2019), Bert: Pre-training of deep bidirectional transformers for language understanding, *em* 'Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)', pp. 4171–4186.

Forster, Yannick, Sebastian Hergeth, Frederik Naujoks, Josef Krems & Andreas Keinath (2019), 'User education in automated driving: Owner's manual and interactive tutorial support mental model formation and human-automation interaction', *Information* **10**(4), 143.

Furth, Sebastian (2018), Linkable Technical Documentation, Tese de doutorado, Universität Würzburg.

Gao, Junyu, Peiyu Liu, Ruoqi Liu, Yuxuan Wang, Chaofan Lin, Huan-Cheng Chang & Cen shao (2024), 'Rethinking chunk size for long-document retrieval', *arXiv preprint arXiv:2505.21700* .

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang & Haofen Wang (2023), 'Retrieval-augmented generation for large language models: A survey', *arXiv preprint arXiv:2312.10997* **2**(1).

Gupta, Aishwarya, Divya Hathwar & A Vijayakumar (2020), 'Introduction to AI chatbots', *International Journal of Engineering Research and Technology* **9**(7), 255–258.

Han, Binglan, Teo Susnjak & Anuradha Mathrani (2024), 'Automating systematic literature reviews with retrieval-augmented generation: A comprehensive overview', *Applied Sciences* **14**(19), 9103.

Han, Yikun, Chunjiang Liu & Pengfei Wang (2023), 'A comprehensive survey on vector database: Storage and retrieval technique, challenge', *arXiv preprint arXiv:2310.11703* .

Heredia Álvaro, José Antonio & Javier González Barreda (2025), 'An advanced retrieval-augmented generation system for manufacturing quality control', *Advanced Engineering Informatics* **64**, 103007.

Huyen, Chip (2025), *AI Engineering*, first[a] edição, O'Reilly Media, Inc., Sebastopol, CA.

Jeong, Cheonsu (2024), 'A study on the implementation method of an agent-based advanced rag system using graph', *arXiv preprint arXiv:2407.19994* .

Julianto, Agung Mulyo Widodo, Gerry Firmansyah & Budi Tjahjono (2024), Large language models using transformers in chat bot based on artificial intelligence, *em* '2024 4th International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)', pp. 391–396.

Jurafsky, Daniel & James H. Martin (2023), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3[a] edição.

Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen & Wen-tau Yih (2020), Dense passage retrieval for open-domain question answering., *em* 'EMNLP (1)', pp. 6769–6781.

Khan, Faizan, Boqi Chen, Daniel Varro & Shane Mcintosh (2021), 'An empirical study of type-related defects in python projects', *IEEE Transactions on Software Engineering* **48**(8), 3145–3158.

Kiciński, Jan, Patryk Chaja, Jan Kiciński & Patryk Chaja (2021), 'Industry 4.0—the fourth industrial revolution', *Climate Change, Human Impact and Green Energy Transformation* pp. 115–140.

Korenčić, Damir, Strahil Ristov, Jelena Repar & Jan Šnajder (2021), 'A topic coverage approach to evaluation of topic models', *IEEE access* **9**, 123280–123312.

Kostric, Ivica & Krisztian Balog (2024), 'A surprisingly simple yet effective multi-query rewriting method for conversational passage retrieval'.

Lane, Hobson & Maria Dyshel (2025), *Natural Language Processing in Action*, second[a] edição, Manning Publications Co., Shelter Island, NY.

LangChain (2024), 'Self-Reflective RAG with LangGraph', Disponível em: https://blog.langchain.dev/agentic-rag-with-langgraph/. Acesso em: 8 de janeiro de 2025.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel et al. (2020), 'Retrieval-augmented generation for knowledge-intensive nlp tasks', *Advances in neural information processing systems* **33**, 9459–9474.

Li, Haitao, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye & Yiqun Liu (2024), 'Llms-as-judges: A comprehensive survey on llm-based evaluation methods'.

Li, Jing, Jingyuan Li, Guo Yang, Lie Yang, Haozhuang Chi & Lichao Yang (2025), 'Applications of large language models and multimodal large models in autonomous driving: A comprehensive review', *Drones* **9**(4).

Li, Zongxi, Zijian Wang, Weiming Wang, Kevin Hung, Haoran Xie & Fu Lee Wang (2025), 'Retrieval-augmented generation for educational application: A systematic survey', *Computers and Education: Artificial Intelligence* p. 100417.

Likert, Rensis (1932), 'A technique for the measurement of attitudes.', *Archives of psychology* .

Lopez-Vega, Henry & Jerker Moodysson (2023), 'Digital Transformation of the Automotive Industry: An Integrating Framework to Analyse Technological Novelty and Breadth', *Industry and Innovation* **30**(1), 67–102.

Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel & Pontus Stenetorp (2022), 'Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity'.

Malin, Ben, Tatiana Kalganova & Nikoloas Boulgouris (2024), 'A review of faithfulness metrics for hallucination assessment in large language models', *arXiv preprint arXiv:2501.00269* .

Manning, Christopher D., Prabhakar Raghavan & Hinrich Schütze (2009), *Introduction to Information Retrieval*, Cambridge University Press.

Mao, Jiageng, Junjie Ye, Yuxi Qian, Marco Pavone & Yue Wang (2023), 'A language agent for autonomous driving', *arXiv preprint arXiv:2311.10813* .

Mars, Mourad (2022), 'From word embeddings to pre-trained language models: A state-of-the-art walkthrough', *Applied Sciences* **12**(17).

Matthias, Karl & Sean P. Kane (2015), *Docker Up & Running*, 1ª edição, O'Reilly Media.

Mavroudis, Vasilios (2024), Langchain.

Maynez, Joshua, Shashi Narayan, Bernd Bohnet & Ryan McDonald (2020), 'On faithfulness and factuality in abstractive summarization'.

Medeiros, Thaís, Morsinaldo Medeiros, Mariana Azevedo, Marianne Silva, Ivanovitch Silva & Daniel G Costa (2023), 'Analysis of language-model-powered Chatbots for query resolution in PDF-based automotive manuals', *Vehicles* **5**(4), 1384–1399.

Mena, Alex Fernando Llerena, Manuel Fernando Gómez Berrezueta, Jerez Mayorga Daniela Alexandra & Peña Pinargote Adolfo Juan (2024), 'Vehicle preventive maintenance: a comprehensive analysis of its impact on society, economic, and environmental factors in general villamil playas city', *South Florida Journal of Development* **5**(1), 65–75.

Microsoft (2024), 'Chunking strategies for vector search', Disponível em: https://learn.microsoft.com/en-us/azure/search/vector-search-how-to-chunk-documents. Acesso em: 5 de Junho de 2025.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado & Jeff Dean (2013), 'Distributed representations of words and phrases and their compositionality', *Advances in neural information processing systems* **26**.

Oduntan, Odunayo E, Ibrahim A Adeyanju, Adeleye S Falohun & Olumide O Obe (2018), 'A comparative analysis of euclidean distance and cosine similarity measure for automated essay-type grading', *Journal of engineering and applied sciences* **13**(11), 4198–4204.

Ogli, Obloev Komronbek Hamza (2024), 'Leveraging pydantic for data validation and settings management in python applications', *MASTERS* **2**(12), 63–69.

OpenAI (2023), 'Gpt-4 technical report', *arXiv preprint arXiv:2303.08774* .

OpenAI (2024*a*), Gpt-40 system card, Relatório técnico, OpenAI.

OpenAI (2024*b*), 'New embedding models and api updates', Disponível em: https://openai.com/blog/new-embedding-models-and-api-updates. Acesso em: 5 de Junho de 2025.

OpenAI (2024*c*), 'Pricing', Disponível em: https://platform.openai.com/docs/pricing. Acesso em: 5 de Junho de 2025.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray et al. (2022), 'Training language models to follow instructions with human feedback', *Advances in neural information processing systems* **35**, 27730–27744.

Pan, James Jie, Jianguo Wang & Guoliang Li (2024), 'Survey of vector database management systems', *The VLDB Journal* **33**(5), 1591–1615.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee & Luke Zettlemoyer (2018), Deep contextualized word representations, *em* M.Walker, H.Ji & A.Stent, eds., 'Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)', Association for Computational Linguistics, New Orleans, Louisiana.

Pryzant, Reid, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu & Michael Zeng (2023), 'Automatic prompt optimization with"gradient descent"and beam search', *arXiv preprint arXiv:2305.03495* .

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever et al. (2018), 'Improving language understanding by generative pre-training'.

Raschka, Sebastian (2025), *Build a Large Language Model from Scratch*, Manning Publications, Shelter Island, NY.

Saleh, Joseph Homer, Archana Tikayat Ray, Katherine S Zhang & Jared S Churchwell (2019), 'Maintenance and inspection as risk factors in helicopter accidents: Analysis and recommendations', *PloS one* **14**(2), e0211424.

Shahade, Aniket K & Priyanka V Deshmukh (2024), Enhancing Natural Language Processing: A Comprehensive Review of Retrieval Augmented Generation, *em* '2024 4th International Conference on Sustainable Expert Systems (ICSES)', IEEE, pp. 609–611.

Shuster, Kurt, Spencer Poff, Moya Chen, Douwe Kiela & Jason Weston (2021), 'Retrieval augmentation reduces hallucination in conversation', *arXiv preprint arXiv:2104.07567* .

Siddharth, L. & Jianxi Luo (2024), 'Retrieval augmented generation using engineering design knowledge', *Knowledge-Based Systems* **303**, 112410.

Sitikhu, Pinky, Kritish Pahi, Pujan Thapa & Subarna Shakya (2019), A comparison of semantic similarity methods for maximum human interpretability, *em* '2019 artificial intelligence for transforming business and society (AITB)', Vol. 1, IEEE, pp. 1–4.

Srivamsi, Dandu, Ommi Mohan Deepak, M.D. Anto Praveena & A. Christy (2023), Cosine Similarity Based Word2Vec Model for Biomedical Data Analysis, *em* '2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)', pp. 1400–1404.

Thapa, Surendrabikram, Shuvam Shiwakoti, Siddhant Bikram Shah, Surabhi Adhikari, Hariram Veeramani, Mehwish Nasim & Usman Naseem (2025), 'Large language models (llm) in computational social science: prospects, current state, and challenges', *Social Network Analysis and Mining* **15**(1), 1–30.

Topsakal, Oguzhan & Tahir Cetin Akinci (2023), Creating large language model applications utilizing langchain: A primer on developing llm apps fast, *em* 'International Conference on Applied Engineering and Natural Sciences', Vol. 1, pp. 1050–1056.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser & Illia Polosukhin (2017), 'Attention is all you need. arxiv preprint arxiv: 170603762'.

Wang, Chen-Kai, Cheng-Rong Ke, Ming-Siang Huang, Inn-Wen Chong, Yi-Hsin Yang, Vincent S Tseng & Hong-Jie Dai (2024), Using Large Language Models for Efficient Cancer Registry Coding in the Real Hospital Setting: A Feasibility Study, *em* 'Biocomputing 2025: Proceedings of the Pacific Symposium', World Scientific, pp. 121–137.

Wang et al. (2024), 'Searching for best practices in retrieval-augmented generation', *arXiv* .

Wang, Jianguo, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu et al. (2021), Milvus: A purpose-built vector data management system, *em* 'Proceedings of the 2021 International Conference on Management of Data', pp. 2614–2627.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le & Denny Zhou (2023), 'Chain-of-thought prompting elicits reasoning in large language models'.

Winkelhake, Uwe (2019), 'Challenges in the digital transformation of the automotive industry', *ATZ worldwide* **121**(7), 36–43.

Woo, Joshua J, Andrew J Yang, Reena J Olsen, Sayyida S Hasan, Danyal H Nawabi, Benedict U Nwachukwu, Riley J Williams III & Prem N Ramkumar (2024), 'Custom large language models improve accuracy: Comparing retrieval augmented generation and artificial intelligence agents to non-custom models for evidence-based medicine', *Arthroscopy: The Journal of Arthroscopic & Related Surgery* .

Wu, Likang, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu et al. (2024), 'A survey on large language models for recommendation', *World Wide Web* **27**(5), 60.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q Weinberger & Yoav Artzi (2019), 'Bertscore: Evaluating text generation with bert', *arXiv preprint arXiv:1904.09675* .

Zhang, Wang, Xin Wang, Qian Wang, Tao Deng & Xiaoru Wu (2024), From text segmentation to enhanced representation learning: A novel approach to multi-label classification for long texts, *em* 'Findings of the Association for Computational Linguistics: EMNLP 2024', pp. 6864–6873.

Zheng, Huaixiu Steven, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le & Denny Zhou (2024), 'Take a step back: Evoking reasoning via abstraction in large language models'.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez & Ion Stoica (2023), 'Judging llm-as-a-judge with mt-bench and chatbot arena'.