UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Cleyton José Rodrigues da Silva

Análise Comparativa de Técnicas de Engenharia de Prompt Aplicadas a Tarefas de Recomendação via LLMs

Recife

2026

Cleyton José Rodrigues da Silva

Análise Comparativa de Técnicas de Engenharia de Prompt Aplicadas a Tarefas de Recomendação via LLMs

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Computação

Curso de Bacharelado em Ciência da Computação

Orientador: Lucas Albertins de Lima

Recife

2026

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Auxiliadora Cunha – CRB-4 1134

S586a Silva, Cleyton José Rodrigues da.
Análise comparativa de técnicas de engenharia de Prompt aplicadas a tarefas de recomendação via LLMs / Cleyton José Rodrigues da Silva. – Recife, 2026.
100 f.; il.

Orientador(a): Lucas Albertins de Lima.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Bacharelado em Ciência da Computação, Recife, BR-PE, 2026.

Inclui referências e apêndice(s).

1. Processamento de Linguagem Natural. 2. Modelos de Linguagem de Grande Escala. 3. Aprendizagem Profunda. 4. Sistemas de Recomendação 5. Engenharia de Prompt. I. Lima, Lucas Albertins de, orient. II. Título

CDD 004

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por CLEYTON JOSÉ RODRIGUES DA SILVA às 15:00 do dia 13/02/2026, na Sala Virtual (https://meet.google.com/bmm-tzui-ygi), como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado "Análise Comparativa de Técnicas de Engenharia de Prompt Aplicadas a Tarefas de Recomendação via LLMs", orientado por LUCAS ALBERTINS DE LIMA e aprovado pela seguinte banca examinadora:

LUCAS ALBERTINS DE LIMA
DC/UFRPE

FILIPE ROLIM CORDEIRO
DC/UFRPE

Dedico aos meus pais, Claudênio e Jacilene, ao meu irmão, Claudênio Filho, e às minhas saudosas avós, Janete e Jandira.

Agradecimentos

Agradeço aos meus pais e irmão, os alicerces do meu apoio moral e emocional, que acreditaram em mim em todos os momentos e fizeram de tudo para me ajudar.

Agradeço ao professor Rinaldo José de Lima por ter me orientado desde o início da minha jornada de estudos no campo de Inteligência Artificial e Processamento de Linguagem Natural, anos atrás, e ao professor Lucas Albertins de Lima por ter sido meu orientador para este TCC.

Por fim, agradeço a todos os professores e secretaria do Departamento de Computação da UFRPE, que contribuíram nesta minha jornada acadêmica, além de ser também agradecido por ter tido a oportunidade de estudar nesta universidade e pelos amigos que nela fiz.

“A primeira e maior vitória é vencer a si mesmo.”
(Platão)

Resumo

Sistemas de recomendação são produzidos para sugerir itens a serem consumidos por usuários clientes de uma determinada plataforma digital, seja em *e-commerce* ou em aplicativos de *streaming* de conteúdo, tendo o intuito de se adequar cada vez mais às preferências dos usuários-alvos, buscando um teor de personalização que contribua para o sucesso da plataforma. As abordagens mais utilizadas atualmente, se apresentam numa combinação de técnicas tradicionais de recomendação com o poder dos LLMs para alavancar a qualidade e precisão das recomendações. Os LLMs são modelos que possuem altas capacidades de compreensão de linguagem e de inferência de linguagem natural e, desde a apresentação do modelo GPT-3, foi evidenciado que o resultado de tarefas executadas por modelos deste tipo podem ter sua qualidade alavancada ao estruturar as *prompts* de interação sob pretextos que melhor extraiam sua capacidade *in-context-learning*. Técnicas de engenharia de prompt como as *zero-shot*, onde se descreve a tarefa em linguagem natural, foram desde então integradas no funcionamento de sistemas de recomendação, e este trabalho analisa uma abordagem onde se usa o LLM como recomendador, propondo uma análise comparativa dos impactos de aplicar quatro variantes de estratégias representativas a técnicas de engenharia de prompt distintas, em substituição a uma estratégia definida como *baseline* de comparação. Experimentos são feitos para diferentes combinações entre três LLMs em duas bases de dados distintas, apresentando resultados experimentais variados dentre as combinações modelo-estratégia, encontrando ganhos de até 17.76% em taxa de acerto de recomendação entre diferentes combinações, com métricas que em parte mostram superação em taxa de acerto contra o *baseline*, e que, por outro lado, mostram o *baseline* mantendo superioridade na qualidade da recomendação gerada.

Palavras-chave: Processamento de Linguagem Natural, Modelos de Linguagem de Grande Escala, Aprendizagem Profunda, Sistemas de Recomendação, Engenharia de Prompt.

Abstract

Recommendation systems are made to predict and suggest items for consumption by client users of virtual platforms, whether on *e-commerce* or on content-streaming applications, while striving to tailor them on user-preference, so that its possible to yield user-personalization that would contribute to platform success. The approaches used by most of current-day systems combine traditional recommendation techniques while leveraging the power of LLMs to further increase recommendation quality and precision. LLMs are modelos that possess high language understanding and natural language inference capabilities, and since the GPT-3 model it was demonstrated that the results given by executed tasks from these types of models, can be of higher quality when leveraging prompt structuring under pretexts that would better extract the model's 'in-context-learning' capacity. Prompt-engineering techniques like zero-shot, where a given task is described by natural language, have since been integrated on recommendation systems, and this work analyses an approach where an LLM is used as a recommender, while proposing a comparative analysis on the impacts of applying four variant strategies that are representative of distinct prompt-engineering techniques, that replace a baseline strategy. Experiments are executed on three different LLMs, for two separate datasets, and the produced results vary among model-estrategy combinations, with findings on up to 17.76% gains in recommendation hit rate, and evaluation metrics showing, on one part, the surpassing on hit rate against baseline results, and on the other part, highlighting the baseline superiority on generated recommendation quality.

Keywords: Natural Language Processing, Large Language Models, Deep-Learning, Recommendation Systems, Prompt Engineering.

Lista de ilustrações

Figura 1 – Gráfico de benchmark de configurações de prompt da publicação do GPT-3 . . . 16
Figura 2 – Desempenho entre inclusão de demonstrações corretas ou nenhuma demonstração na prompt . . . 16
Figura 3 – Listagem de preços de utilização das APIs dos modelos GPT . . . 17
Figura 4 – Arquitetura do modelo Transformer . . . 21
Figura 5 – Exemplos de objetivos de treinamentos não-supervisionados . . . 23
Figura 6 – Procedimentos de treinamento o modelo BERT . . . 24
Figura 7 – Arquitetura do modelo GPT-1 . . . 24
Figura 8 – Desempenho *in-context learning* por número de parâmetros . . . 25
Figura 9 – Exemplos das técnicas de engenharia de prompt . . . 26
Figura 10 – Exemplos das técnicas de engenharia de prompt . . . 27
Figura 11 – Linha do tempo das técnicas de recomendação . . . 28
Figura 12 – Métodos de recomendação com LLMs . . . 29
Figura 13 – Arquitetura do sistema de recomendação de Wang e Lim (2023) . . . 39
Figura 14 – Arquitetura do sistema de recomendação . . . 42
Figura 15 – Esquema do pré-processamento . . . 52
Figura 16 – Exemplo de matriz usuario-item . . . 54
Figura 17 – Comparação das estratégias nos subconjuntos do dataset de análises Amazon . . . 63
Figura 18 – Comparação de resultados no dataset MovieLens 100K . . . 68
Figura 19 – Matriz *user-user* de coocorrências de interações em itens entre usuários . . . 98

Lista de tabelas

Tabela 1 – Resumo dos trabalhos relacionados 37
Tabela 2 – Resultados dos experimentos na base de dados Amazon Reviews Dataset com o modelo GPT-3.5-Turbo-Instruct 61
Tabela 3 – Resultados dos experimentos na base de dados Amazon Reviews Dataset com o modelo Llama-3.1 8B Instruct 62
Tabela 4 – Resultados dos experimentos na base de dados Amazon Reviews Dataset com o modelo Gemma-3-4B-IT 62
Tabela 5 – *Ablation study* dos aspectos das prompts com melhores resultados das execuções para o *subset* Amazon Luxury Beauty 65
Tabela 6 – Resultados dos experimentos na base de dados MovieLens 100K em comparação com sistemas externos 67
Tabela 7 – Custos de utilização do GPT-3.5-Turbo-Instruct 69
Tabela 8 – Exemplo de 15 maiores somatórios dos pesos de itens candidatos para um usuário-alvo 97

Lista de abreviaturas e siglas

API	Application Programming Interface
GAT	Graph Attention Networks
LLM	Large Language Model
PLM	Pre-trained Language Model
PLN	Processamento de Linguagem Natural
RNN	Recurrent Neural Networks
SR	Sistema de Recomendação

Sumário

Lista de ilustrações 9

1 INTRODUÇÃO 14
1.1 Problema de Pesquisa 15
1.2 Contribuições 18
1.3 Estrutura do trabalho 19

2 FUNDAMENTAÇÃO TEÓRICA 20
2.1 Processamento de Linguagem Natural 20
2.2 O modelo *Transformer* e o mecanismo *Self-Attention* 21
2.3 Large language models 22
2.4 Engenharia de prompts 25
2.5 Sistemas de Recomendação 27

3 TRABALHOS RELACIONADOS 30
3.1 Recomendação com modelo *Transformer* pré-LLMs 30
3.2 Recomendação com LLMs baseado na técnica *Zero-Shot* 31
3.3 Ajuste Fino de LLMs para Recomendação 33
3.4 Semelhanças e Diferenças com o Sistema Proposto 36
3.5 Conclusão 36

4 PROPOSTA 38
4.1 Modificações no *framework* do sistema de recomendação 41
4.2 Técnicas de engenharia de prompt utilizadas 43
4.2.1 A estratégia do baseline: *Three-step prompting* 44
4.2.2 *Two-step prompting* 44
4.2.3 *Single-step prompting* 45
4.2.4 *Chain-of-thoughts prompting* 45
4.2.5 *One-shot prompting* 45
4.3 Modelos LLM escolhidos 47
4.4 Conclusão 47

5 EXPERIMENTOS 49
5.1 Configuração Experimental 49
5.1.1 Bases de Dados 49
5.1.2 Ambiente de execução dos experimentos 49
5.1.3 Métricas de Avaliação de Desempenho 50

5.1.3.1 *Hit Rate* . . . 50
5.1.3.2 *Normalized Distributed Cumulative Gain* (NDCG) . . . 50
5.1.3.3 *Recall* . . . 51
5.2 Criação das amostras . . . **52**
5.2.1 Pré-processamento de Dados . . . 52
5.2.2 Filtragem por usuários . . . 54
5.2.3 Definição de item *Ground-Truth* . . . 57
5.2.4 Formatação dos templates das estratégias . . . 58
5.3 Resultados e Análise . . . **60**
5.3.1 Resultados dos experimentos no Amazon Reviews Dataset . . . 60
5.3.2 *Ablation Study*: Impacto dos aspectos das prompts . . . 63
5.3.3 Comparação com trabalhos externos . . . 66
5.3.4 Custos computacionais . . . 69
5.4 Conclusão . . . **70**

6 CONCLUSÃO . . . **73**
6.1 Limitações . . . **74**
6.2 Trabalhos Futuros . . . **75**

REFERÊNCIAS . . . **76**

A APÊNDICE . . . **82**
A.1 Prompts de entrada das estratégias aplicadas para recomendação **82**
A.1.1 Prompts das estratégias *Zero-shot* . . . 82
A.1.2 Prompt da estratégia *Chain-of-Thoughts* . . . 86
A.1.3 Prompts das estratégias *One-shot* . . . 88
A.2 Exemplos de artefatos da filtragem por usuários . . . **97**
A.3 Snippets . . . **99**

1 Introdução

De acordo com Raza et al. (2026), sistemas de recomendação são aqueles produzidos para prever e sugerir itens de consumo para usuários que são clientes de um negócio digital definido, seja em produtos num *e-commerce* para livros e artigos de beleza, ou em plataformas para itens não palpáveis como músicas e filmes. A intenção de um sistema de recomendação é de que a previsão seja feita por uma personalização criada sob medida dos interesses do cliente a quem se vai recomendar. Raza et al. (2026) ainda citam que, tamanha é a importância destes sistemas, que empresas tornadas hoje corporações gigantes, vieram a se consolidar a partir da integração dos SRs em seu negócio, com empresas como a Amazon relatando que 35% de sua receita vem de seu SR, enquanto que a Netflix relata rendimentos de até $33 Bilhões atribuídos à retenção de clientes permitida pelo sucesso seu SR.

O progresso da pesquisa dos sistemas de recomendação se deu, por um tempo, em paralelo ao progresso da pesquisa de *deep-learning* e da abordagem dos desafios de Processamento de Linguagem Natural (GRBOVIC et al., 2015; HIRSCHBERG; MANNING, 2015), mas logo passou-se a adotar métodos que integrassem os modelos de redes neurais artificiais (SHENBIN et al., 2020; HAMILTON; YING; LESKOVEC, 2017; RAZA et al., 2026). No passar do tempo, técnicas tradicionais para SRs, como a filtragem colaborativa — baseada na previsão de preferências de um usuário-alvo — foram integradas em sistemas híbridos de recomendação, que as combinavam com modelos de aprendizagem de máquina para a decisão da recomendação. Dali em diante, visto que métodos anteriores tinham dificuldades para maior extrair maior nuance das relações item-usuário e de definir personalizações mais complexas, novas abordagens para SRs iriam se basear na ultilização do modelos do estado da arte da aprendizagem profunda, sendo propostos sistemas com *Multi-Layer Perceptrons*, Redes Neurais Convolucionais e Recorrentes, buscando-se melhorar a precisão e o nível de personalização de tais SRs (RAZA et al., 2026).

Desenvolvidos a partir da abordagem de desafios de Processamento de Linguagem Natural como a Tradução Automática e a Compreensão de Linguagem Natural, os LLMs (modelos de linguagem de grande escala) demonstraram grande poder de reconhecimento de padrões e compreensão de contexto (RADFORD et al., 2018), e à medida que a pesquisa de sistemas de recomendação acompanhava os avanços dos estudos de *deep-learning*, os LLMs foram integradas para permitir ainda maior personalização na recomendação (PENG et al., 2025). Essa melhoria se deu graças as capacidades de aprendizagem em contexto (*In-Context-Learning*) dos LLMs, onde o modelo reconhece e interpreta relações de preferências do usuário em tempo de

inferência, sem necessidade, a priori, de novo treinamento. Com isso, desde então os SR integram as LLMs usando do seu poder de adaptabilidade para alavancar a qualidade de recomendações geradas, a partir de prompts personalizadas e também usando o poder de processamento de linguagem natural desses modelos para propostas de SRs conversacionais ou ainda via operações *zero-shot*, onde a tarefa é descrita diretamente por linguagem natural (PENG et al., 2025; RAZA et al., 2026).

Wang e Lim (2023) apresentam um SR que combina a tradicional técnica de filtragem colaborativa com uma variante do LLM GPT-3.5 para recomendação, sob estratégia de montagem de prompt batizada *three-step-zero-shot*. Na arquitetura lógica deste sistema, primeiramente usa-se a filtragem colaborativa para produzir uma seleção de itens candidatos de recomendação, a partir de uma aproximação das preferências do usuário alvo de recomendação. Esta seleção é então utilizada para executar a estratégia *three-step-zero-shot*, que se trata da aplicação da técnica *zero-shot* de engenharia de prompt, onde, via duas prompts sequênciais, constrói-se um contexto que contenha as preferências do usuário a ser incluido numa terceira prompt que solicita a inferência de recomendação de dez itens.

Apesar de promover bons resultados de recomendação, algo a destacar é o quão custosa a abordagem desta estratégia pode ser, visto que, após a filtragem colaborativa que avalia cada usuário-alvo entre todos outros usuários, na perspectiva de utilização do LLM a recomendação se dá, para cada usuário-alvo, por três prompts que depois de formatadas podem compor um número considerável de *tokens* a serem processados pelo modelo. Diante disso, identifica-se que esta estratégia custosa poderia ser substituida, visto que o *framework* original dá a possibilidade de integração de novas estratégias de prompt, que apliquem outras técnicas de *prompt-engineering*, para buscar resultados comparáveis a estratégia original de maneira mais eficiente.

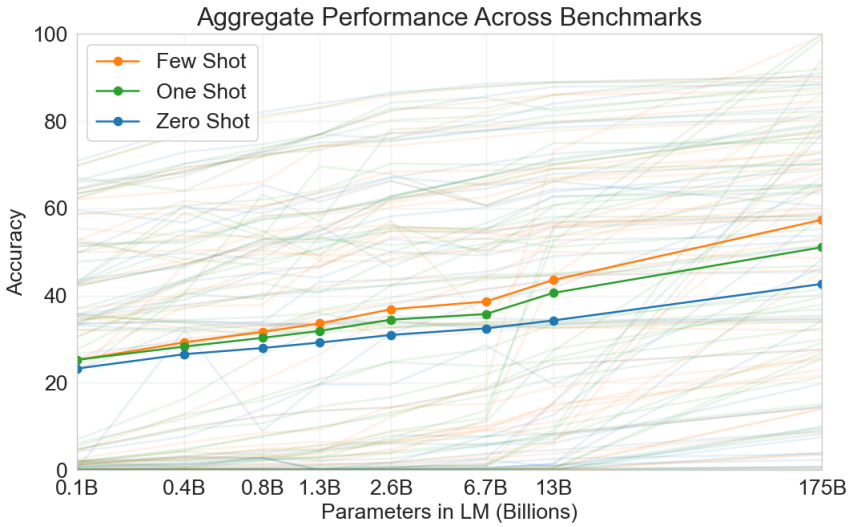
1.1 Problema de Pesquisa

A publicação do modelo GPT-3 trouxe grande notoriedade a si mesmo devido ao demonstrado potencial de realização de tarefas inúmeras, quando descritas em linguagem natural, onde foi evidenciado que os resultados de tais tarefas podem ter qualidade alavancada quando se interage com o modelo de maneiras textuais específicas, definidas para a *prompt* de interação com o LLM (BROWN et al., 2020). A partir disso, os pretextos para estruturação de tarefas para modelos como o GPT-3 passaram a constituir o estudo da engenharia de prompt.

Com as técnicas fundamentais *zero-shot, one-shot e few-shots* sendo analisadas juntamente à publicação do GPT-3, ilustrado na figura 1, foi demonstrado que o desempenho destas três estratégias tendenciavam, até então, a ser de graus diferen-

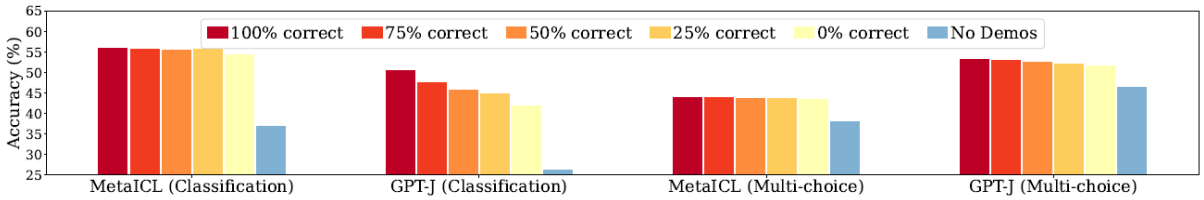
tes devido à suas características de inclusão de demonstração (ou exemplo) da tarefa a ser executada na prompt que a solicita — a técnica *zero-shot* apenas descrevendo em linguagem natural, a *one-shot* incluindo um exemplo e a *few-shots* incluindo mais de um exemplo — com a técnica *few-shots* sendo superior naquela ocasião. Anos depois, Min et al. (2022) demonstram que, apesar das descobertas de Brown et al. (2020) se confirmarem em seu estudo, ainda é possível extrair da capacidade *in-context-learning* dos LLMs resultados de precisão comparáveis ao usar a técnica *zero-shot* que inclui nenhuma demonstração, como mostrado na figura 2.

Figura 1 – Gráfico de benchmark de configurações de prompt da publicação do GPT-3



Fonte: Brown et al. (2020)

Figura 2 – Desempenho entre inclusão de demonstrações corretas ou nenhuma demonstração na prompt

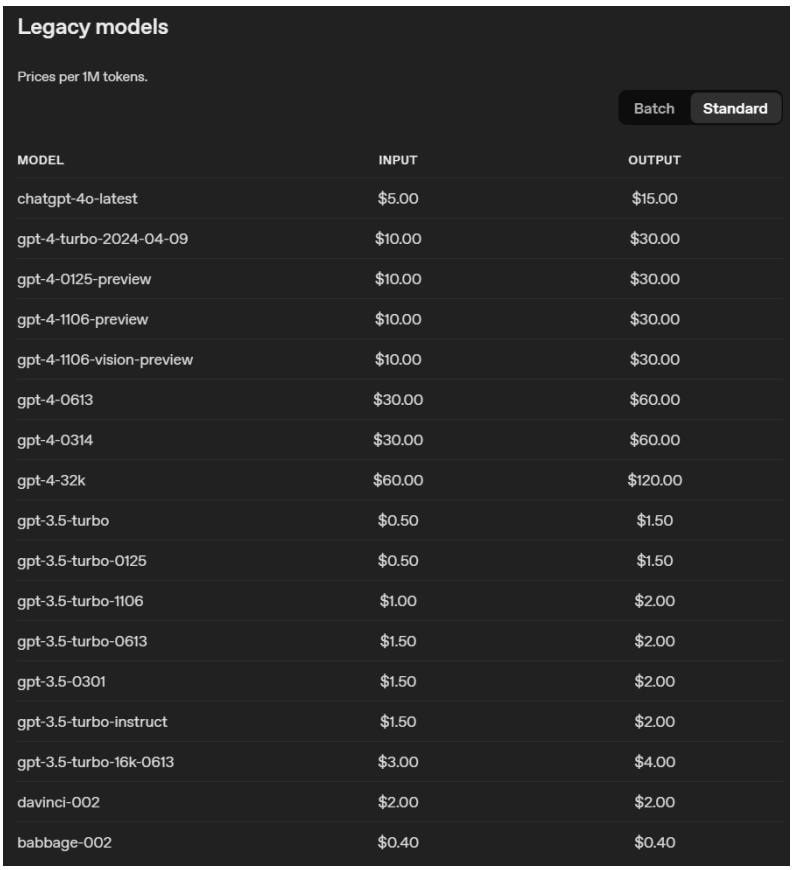


Fonte: Min et al. (2022)

Analisando a perspectiva do uso de um modelo como o GPT-3, é perceptível que os custos cobrados por processamento de *tokens* de entrada e saída gerada, ilustrados na figura 3, podem representar um empedimento para sua adoção. Apesar de serem inicialmente irrisórios em usos especificamente pontuais, cobrando-se poucos centávos por token processado, quando aplicados numa operação de larga escala onde seriam realizadas muitas requisições, em longos e contínuos períodos de tempo — como seria o caso de um sistema de recomendação aplicado para uma plataforma

digital hipotética, de funcionamento contínuo, processando múltiplas informações de itens para múltiplos usuários — estes custos se traduziriam eventualmente em valores exorbitantes.

Figura 3 – Listagem de preços de utilização das APIs dos modelos GPT



Legacy models

Prices per 1M tokens.

Batch | **Standard**

MODEL	INPUT	OUTPUT
chatgpt-4o-latest	$5.00	$15.00
gpt-4-turbo-2024-04-09	$10.00	$30.00
gpt-4-0125-preview	$10.00	$30.00
gpt-4-1106-preview	$10.00	$30.00
gpt-4-1106-vision-preview	$10.00	$30.00
gpt-4-0613	$30.00	$60.00
gpt-4-0314	$30.00	$60.00
gpt-4-32k	$60.00	$120.00
gpt-3.5-turbo	$0.50	$1.50
gpt-3.5-turbo-0125	$0.50	$1.50
gpt-3.5-turbo-1106	$1.00	$2.00
gpt-3.5-turbo-0613	$1.50	$2.00
gpt-3.5-0301	$1.50	$2.00
gpt-3.5-turbo-instruct	$1.50	$2.00
gpt-3.5-turbo-16k-0613	$3.00	$4.00
davinci-002	$2.00	$2.00
babbage-002	$0.40	$0.40

Fonte: (OPENAI, 2026)

Diante disso e percendo a viabilidade das técnicas de engenharia de prompt em extrair níveis semelhantes da capacidade *in-context-learning* dos LLMs, mesmo diferindo em aspectos que as tornariam teoricamente mais “fracas”, levantam-se os seguintes problemas:

- ***Como técnicas de engenharia de prompt podem melhorar o desempenho da recomendação via LLMs, ao diferirem na forma em que extraem níveis diferentes do poder de aprendizagem em contexto (In-Context-Learning) de tais modelos?***
- ***Como LLMs mais acessíveis podem substituir modelos robustos e pagos, mantendo o desempenho de recomendação?***

O presente trabalho visa conduzir uma análise comparativa do impacto da aplicação de técnicas distintas de engenharia de prompt na tarefa de recomendação via

LLMs, ao propor uma melhoria para o *framework* do sistema de recomendação apresentado por Wang e Lim (2023), em forma da modificação do componente de engenharia de prompt do sistema, por quatro variantes de estratégias apresentadas por este trabalho, modelas para serem representativas de cada técnica de engenharia de prompt, com a finalidade de que seja analisado seus desempenhos diante da estratégia original, que é definida como ***baseline*** de comparação. O desempenho do LLM usando cada estratégia de prompt será medido pelas métricas avaliativas *HitRate*, *NDCG* e *Recall*, e o impacto que cada estratégia têm no desempenho da recomendação é avaliado a partir de uma ótica que individualiza os aspectos característicos de cada prompt dentre as estratégias propostas, buscando analisa-los como fatores de causalidade da performance. Propõe-se ainda uma face comparativa que inclui a avaliação do desempenho da tarefa de recomendação quando feita via LLMs *open-source* e de robustez reduzida comparada ao modelo da OpenAI — robustez identificada, a priori, em número de parâmetros.

1.2 Contribuições

Como forma de alcançar o objetivo geral proposto, este trabalho produziu as seguintes contribuições:

1. **Proposta de novas estratégias de prompt para o *framework* do Sistema de Recomendação:** Foram propostas quatro estratégias de prompt buscando superar o desempenho da estratégia *baseline* na tarefa de recomendação via LLMs, sendo estratégias representativas de técnicas distintas de engenharia de prompt, nomeadamente as técnicas *zero-shot*, *one-shot* e *chain-of-thoughts*.

2. **Implementação do *framework* modificado:** Este trabalho implementa o *framework* do SR de Wang e Lim (2023), modificando-o no sentido da substituição da estratégia de prompt original, para aplicação das estratégias de prompt propostas.

3. **Avaliação comparativa das estratégias propostas:** Este trabalho avalia comparativamente o desempenho das estratégias de prompt propostas contra a estratégia *baseline* na tarefa de recomendação, nas duas base de dados definidas — Amazon Reviews Dataset, uma escolha desse trabalho, e MovieLens 100K, a base originalmente avaliada pelo *baseline*.

4. **Comparação com trabalhos de SR externos:** Realiza-se uma comparação de desempenho do *baseline* e das estratégias propostas com sistemas de recomendação provindos de trabalhos externos, sendo avaliados na base de dados MovieLens 100K.

5. **Avaliação de LLMs menos robustos e *open-source* para o Sistema de Recomendação**: Analisa-se a capacidade na tarefa de recomendação de LLMs de robustez reduzida, mas modelados sob ajuste fino na operação *instruction-tuning*, de forma a comparar o seu desempenho diante do modelo GPT-3.5-Turbo-Instruct, aplicando as mesmas estratégias de prompt para as duas bases de dados — foram escolhidas os modelos Llama 3.1 8B Instruct e Gemma-3-4B-IT.

1.3 Estrutura do trabalho

Este trabalho está estruturado em seis capítulos, apresentando a pesquisa em sua fundamentação, desenvolvimento da proposta e resultados comparativos.

Este que é o **Capítulo 1**, introduz o tema de sistemas de recomendação com LLMs, contextualizando sobre o progresso das tendências de adoção desses modelos, apresentando brevemente o sistema que é objeto de estudo do trabalho. Também apresenta-se o problema de pesquisa abordado, o objetivo geral e as contribuições deste trabalho.

O **Capítulo 2** apresenta os conceitos dos temas envolvidos por este trabalho, contextualizando sobre as áreas de estudo de Processamento de Linguagem Natural, LLMs, Engenharia de Prompt e Sistemas de Recomendação.

O **Capítulo 3** analisa os trabalhos que se relacionam com este trabalho, discutindo outras propostas de Sistemas de Recomendação que adotaram o uso de técnicas tradicionais de recomendação, Aprendizagem Profunda e LLMs.

O **Capítulo 4** apresenta em detalhes a proposta de modificação no *framework* original da estratégia *baseline* de recomendação, explicando sobre as quatro propostas de estratégias de prompt e as arquiteturas dos sistemas avaliados.

O **Capítulo 5** apresenta os detalhes de implementação dos experimentos de avaliação da proposta, os resultados obtidos entre as diferentes configurações juntamente a análise sobre eles, um estudo de ablação sobre as estratégias de prompt propostas e uma conclusão que discute sobre os experimentos.

Por fim, o **Capítulo 6** apresenta a conclusão do trabalho, sumarizando os resultados obtidos pelos experimentos, discutindo se eles validam as hipóteses estabelecidas e apresentando as limitações do trabalho.

2 Fundamentação Teórica

Neste capítulo são apresentados conceitos acerca dos temas principais abordados por este trabalho, colocados, além disso, como forma de contextualização das áreas de estudo envolvidas.

2.1 Processamento de Linguagem Natural

Segundo (HIRSCHBERG; MANNING, 2015) o estudo de Linguística Computacional, também conhecido como Processamento de Linguagem Natural refere-se ao campo da ciência da computação que abrange técnicas computacionais para aprendizado, compreensão e produção de conteúdo de linguagem humana. Entre os primeiros trabalhos que envolveram a ideia de processamento de linguagem natural, nos anos 1950, está o trabalho de Alan Turing, que propôs sobre a possibilidade de uma máquina ter a capacidade de compreender e raciocinar a língua humana, a ser avaliada no Teste de Turing (TURING, 1950; JURAFSKY; MARTIN, 2009).

Ao longo das décadas seguintes, foram desenvolvidas propostas que viriam a contribuir para o estado da arte da construção de modelos de linguagem, como é o caso dos *Vector Space Models* (VSMs) e *Word-Embeddings* (Ghaseminejad Raeini, 2025). Os VSMs são modelos de representação, onde palavras são definidas como vetores de característica num espaço vetorial e, a partir deste conceito *Word-Embeddings* se tornariam modelos estatísticos para geração desses vetores. O modelo *Word2Vec* (MIKOLOV et al., 2013) é um modelo produzido para prever palavras que seriam sequências de outras, que servem como ponto de partida. Isso é possível com treinamentos que adaptam a previsão de uma palavra, dado o contexto de proximidade que ela tem com outras palavras, isto é, quais palavras aparecem antes ou depois de outras (Ghaseminejad Raeini, 2025).

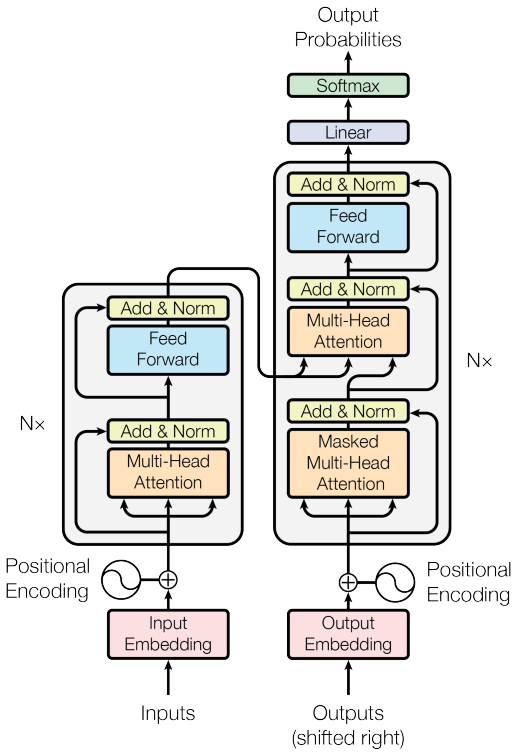
Mais recentemente, as tarefas de PLN passaram a ser estudadas sob a perspectiva do uso dos modelos de linguagem pré-treinados (PLMs), uma vez que estes novos modelos passaram a possibilitar a construção de representações de texto em escalas muito mais robustas, o que é de grande utilidade para abordar tarefas como a Inferência de Linguagem Natural (SCHICK; SCHüTZE, 2021) e a Compreensão de Linguagem Natural (DEVLIN et al., 2019), com novos níveis de complexidade para a realização de tais tarefas sendo desenvolvidos.

2.2 O modelo *Transformer* e o mecanismo *Self-Attention*

Ao longo das décadas após o reaquecimento do campo de estudo das redes neurais artificiais nos anos 1980 (SCHMIDHUBER, 2015), foram dados muitos frutos de pesquisas, nas formas de modelos como as *Convolutional Neural Networks* (Redes Neurais Convolucionais) e *Recurrent Neural Networks* (Redes Neurais Recorrentes), explorados assíduamente em seu tempo, tornados hoje parte do estado da arte do campo de pesquisa de *Deep-Learning*. A partir do conceito das RNNs, foi baseado o modelo de gigante importância para o progresso do campo de estudo da Inteligência Artificial, o modelo *Transformer*.

A arquitetura *Transformer*, mostrada na figura 4, foi uma inovação dentre os modelos de redes neurais artificiais, destinado, inicialmente em sua concepção, para tarefas relacionadas ao processamento de linguagem natural. Conforme se constata na sua origem por Vaswani et al. (2017), o modelo inicialmente aplicado na tarefa da tradução automática, proporcionou principalmente a melhor e mais eficiente aprendizagem do contexto pertencente a cada palavra das sentenças processadas, o que foi de grande importância para alcançar maior precisão na classificação dos significados de cada frase. Isto significou que, dado um treinamento eficiente do modelo na tarefa de tradução de frases, ao levar em conta os fatores de ordenação e contexto de cada palavra, era possível a tradução mais acurada do significado das frases, com resultado longe de ser apenas um conjunto de traduções diretas mapeadas da frase original.

Figura 4 – Arquitetura do modelo Transformer



Fonte: Vaswani et al. (2017)

Isto foi feito com o uso do chamado *Positional Encoding* e o mecanismo nomeado como *Self-Attention*, uma nova abordagem do mecanismo *Attention*, originalmente utilizado num modelo de redes neurais recorrentes para tradução automática (BAHDANAU; CHO; BENGIO, 2016). O Attention consiste numa estrutura onde, à medida que as entradas (textos) iteram no modelo, para cada palavra num texto de entrada, são levadas em conta as palavras que estão ao seu redor, computando, por palavra, somados vetores de contexto que condicionam a nível de treinamento maiores pesos a probabilidade de uma tradução, que por sua vez acaba sendo a mais precisa e a resultante.

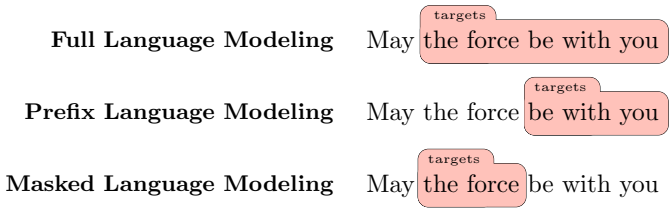
Numa ideia semelhante, dado uma sentença de entrada, o Self-Attention leva em conta a posição de cada palavra, mas analisa seu contexto ao rastreá-las, no próprio texto de entrada, pela posição em que cada uma se encontra, relacionando cada posicionamento a todas as palavras antes de si, para computar este rastreamento em uma representação da sequência que é repassada para treino. Desta forma, ao dispensar a limitação antes encontrada nas rede neurais recorrentes (*RNNs*), em restringir à entrada sequencial das sentenças (BAHDANAU; CHO; BENGIO, 2016), a possibilidade da entrada paralelizada de dados no modelo original permitiu que a Transformer fosse treinada de maneira mais eficiente, e possibilitou o potencial de aprendizagem de um nível muito maior de contextos das palavras, inclusive em textos mais extensos (VASWANI et al., 2017).

2.3 Large language models

O modelo *Transformer* é definido hoje como um marco na pesquisa de modelos de linguagem e *deep-learning* e, a partir dele, foram apresentados os *Large Language Models* (modelos de linguagem de grande escala), que representaram novos marcos no campo de estudo (MINAEE et al., 2025).

Um LLM, essencialmente, trata-se de um modelo de rede neural artificial, baseada numa arquitetura *Transformer,* pré-treinado sob um objetivo específico de treinamento não-supervisionado, como aqueles da figura 5 a partir de técnicas que determinam a estrutura de predição dele (NAVEED et al., 2025). Estes modelos, pré-treinados com uma quantidade massivamente grande de dados em linguagem natural, submetidos a operações de ajuste fino em seguintes etapas de modelagem, se constituem de uma quantidade proporcionalmente massiva de parâmetros aprendidos, este fator permite a eles a capacidade de predizer as respostas em linguagem natural, estruturadas de forma altamente acurada em relação ao que se espera de um texto humanamente inteligível (YU et al., 2024; NAVEED et al., 2025).

Figura 5 – Exemplos de objetivos de treinamentos não-supervisionados



Fonte: Wang et al. (2022)

A tarefa de interpretação de linguagens, ou a tradução de textos por meio de predição num modelo transformer, foi o exemplo que demonstrou tal robustez no aprendizado e predição de sequências de palavras. O mecanismo de Atenção junto à Arquitetura Transformer, foi um catalizador para maiores avanços no campo de modelos de linguagem, culminando em modelos como o BERT (*Bidirectional Encoder Representations from Transformers*) e o GPT-1 (onde acrônimo seria para *Generative Pre-trained Transformer*), cuja arquitetura ganhou alto destaque no mundo (MINAEE et al., 2025).

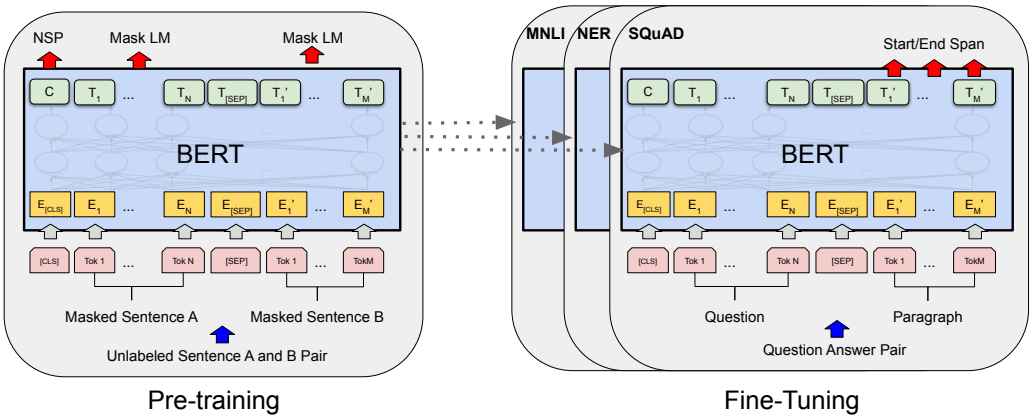
Minaee et al. (2025) cita alguns dos principais modelos de linguagem pré-treinados — *Pre-trained Language Models* ou PLM — baseados em *Transformers*, de onde iriam surgir os LLMs:

- **PLMs *Encoder-only***: originalmente propostos para tarefas como classificação de texto, eles consistem de redes de codificadores, como o BERT (Birectional Encoder Representations from Transformers). o BERT consiste de três módulos:

  1. um módulo de embedding que converte textos de entrada em sequências de vetores embedding;
  2. uma pilha de codificadores de Transformadores que converte os vetores embedding em vetores de representação contextual;
  3. uma camada *dense* (rede inteiramente conectada, MLPs) que converte os vetores de representação em vetores *one-hot* (um valor no espaço do vetor por token ou representação)

  Dada a estrutura, num procedimento ilustrado pela figura 6, o BERT realiza o pre-treinamento usando o modelo de mascaramento de linguagem nos textos de treinamento (mascaramento de tokens para predição pela rede neural) e a predição de sentença seguinte. Pré-treinado, o BERT pode ser submetido a ajuste fino com a adição de camada extra de classificação para tarefas de compreensão de linguagem, como classificação (DEVLIN et al., 2019).

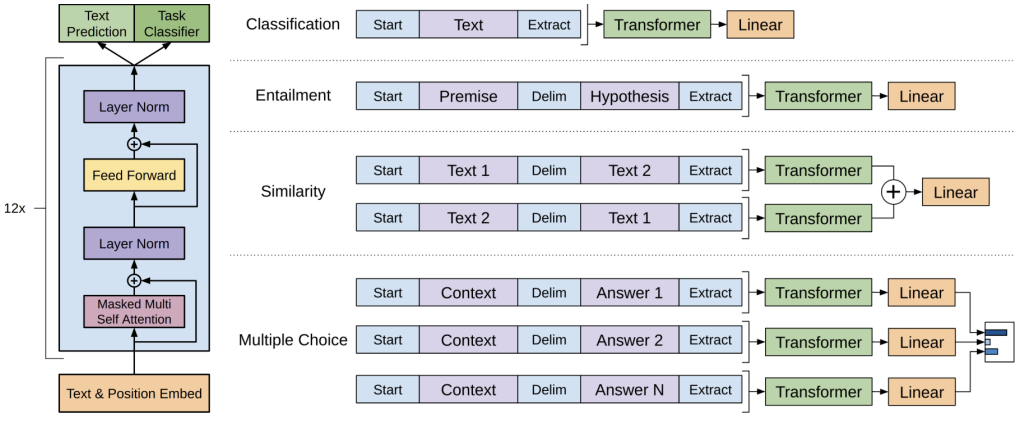
Figura 6 – Procedimentos de treinamento o modelo BERT



Fonte: Devlin et al. (2019)

- **PLMs *Decoder-only*:** Dois dos modelos de PLM *Decoder-only*, são o GPT-1 e GPT-2, pela OpenAI. Baseado na arquitetura do modelo *Transformer* original, o GPT-1 se trata de um modelo *decoder-only* feito por pré-treinamento em cima de um diverso corpo de texto não rotulado, sob aprendizado auto-supervisionado (na predição de palavra/token seguinte), seguido de ajuste fino numa tarefa específica usando volume menor de amostras de treino (RADFORD et al., 2018). O pré-treinamento foi feito em representações de conteúdos de livros dadas pelo dataset *BooksCorpus* (citação), para que o modelo "aprendesse" relações de informação *long-range*, de longo alcance, dadas as sequências contínuas de texto encontrado num livro. A operação de ajuste fino supervisionado se deu no treinamento para realização de tarefas de compreensão de linguagem, como a tarefa de Inferência de Linguagem Natural — onde se identifica o vínculo textual entre duas sentenças, julgando entre *entailment* (vínculo), *contradiction* (contradição) e *neutral* (neutro) — e Similaridade semântica — inferindo se duas sentenças são semanticamente equivalentes (RADFORD et al., 2018). Sua arquitetura e treinamento são ilustados na figura 7.

Figura 7 – Arquitetura do modelo GPT-1



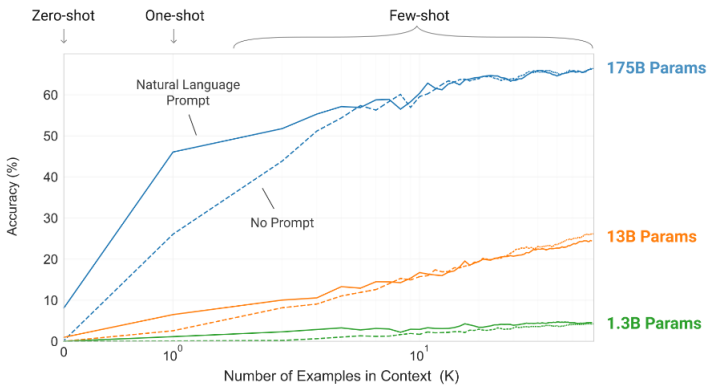
Fonte: Radford et al. (2018)

O modelo GPT-3 é considerado por muitos o primeiro LLM, devido ambos a sua escala de parâmetros maior que outros PLMs e ao grande poder em executar diversas tarefas de PLN, como as tarefas de Tradução e *Question-Answering*, além de tarefas que exigiam "compreensão" do contexto fornecido para geração da resposta (MINAEE et al., 2025). Nesse meio tempo, foram apresentadas novas "famílias" de LLMs desenvolvidos por organização alheias à OpenAI, como os modelos LLaMa da Meta e PaLM da Google, seguindo os pretextos fundamentais das redes *Transformers* para trazer abordagens variadas em seu desenvolvimento (TOUVRON et al., 2023; CHOWDHERY et al., 2022).

2.4 Engenharia de prompts

Os LLMs conseguem, em tempo de inferência, identificar qual tarefa está sendo solicitada na prompt que representa o ponto de partida da geração a ser efetuada. Brown et al. (2020) chamam essa capacidade de *in-context-learning*, possibilitada devido às amplas capacidades de reconhecimento de padrões desenvolvidas no pretreinamento não-supervisionado do modelo. A capacidade *in-context-learning* do modelo é melhor induzida para geração quando a prompt é estruturada de maneiras específicas, de forma que contenha elementos textuais que contribuam para o reconhecimento dos padrões que levam a predições mais acuradas do que devia ser o texto inferido, dada a tarefa solicitada (BROWN et al., 2020).

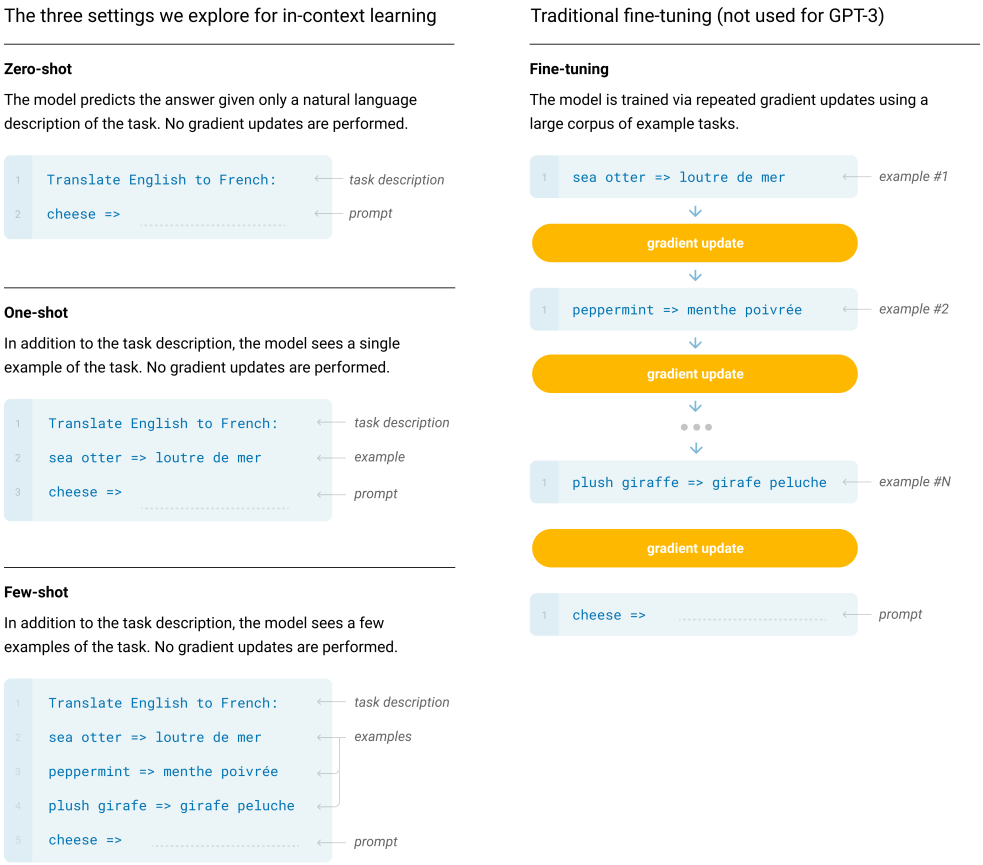
Figura 8 – Desempenho *in-context learning* por número de parâmetros



Fonte: Brown et al. (2020)

As técnicas que envolvem pretextos específicos para estruturações de prompts, definem o que se envolve no estudo da engenharia de prompts, sendo colocadas em voga juntamente a publicação do modelo GPT-3, que explorou tais possibilidades de estruturação de prompt em volta do constatação da capacidade *in-context-learning* — colocadas em comparação na figura 8 — onde não se necessita, a priori, uma continuação de treinamento do modelo (BROWN et al., 2020; CHEN et al., 2025).

Figura 9 – Exemplos das técnicas de engenharia de prompt



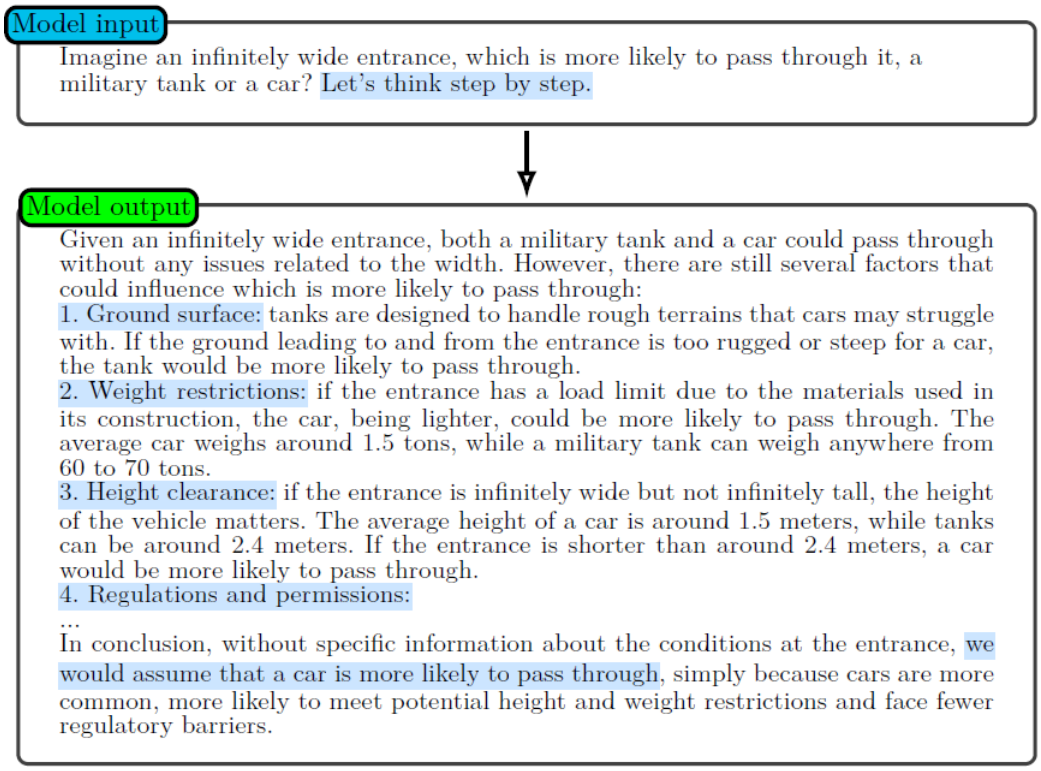
Fonte: Brown et al. (2020)

Algumas entre as principais técnicas de engenharia de prompt, com exemplos ilustrados nas figuras 9 e 10 estão:

- ***Role-prompting***: a prompt é construída de forma a instruir o modelo a assumir um papel sob qual o ponto de vista deve ser contexto para geração. Por exemplo, pode-se definir o papel de historiador e solicitar informações históricas (CHEN et al., 2025).
- **Zero-shot:** os modelos de linguagem de grande escala são, por concepção, próprios para o estilo de prompts *zero-shot*, onde não se demonstra nenhum exemplo de execução da tarefa solicitada, a ser definida como uma instrução em forma de linguagem natural.(NAVEED et al., 2025).
- **One-shot e Few-shots:** São duas técnicas que envolvem incluir, na prompt, demonstrações de como a tarefa deve ser efetuada. A técnica one-shot tem esse nome pois nela se inclui apenas um exemplo, onde a técnica few-shots inclui mais de uma demonstração (BROWN et al., 2020).

- **Chain-of-thoughts:** se trata da inclusão de passos de raciocínio intermédiarios para guiar a geração do LLM, que toma cada "raciocínio" inferido ou fornecido como lógica motivadora da geração (CHEN et al., 2025).

Figura 10 – Exemplos das técnicas de engenharia de prompt



Fonte: Chen et al. (2025)

2.5 Sistemas de Recomendação

Segundo (LI et al., 2023), sistemas de recomendação são aqueles projetados para oferecer recomendações personalizadas de produtos e serviços — sejam livros, música, filmes, entre outros — para usuários, de maneira a adequar a experiência virtual do usuário cliente e, o fator principal dos sistemas de recomendação é de sugerir itens relevantes para os usuários, com base na aplicação de técnicas para identificação de suas preferências, de características de seus itens e de suas interações. O contexto dos itens de recomendação é variado, abrangendo plataformas de *e-commerce*, notícias, finanças até o contexto do entretenimento, onde se aplicaram abordagens diferentes para recomendação no decorrer dos anos (GRBOVIC et al., 2015; WU et al., 2022).

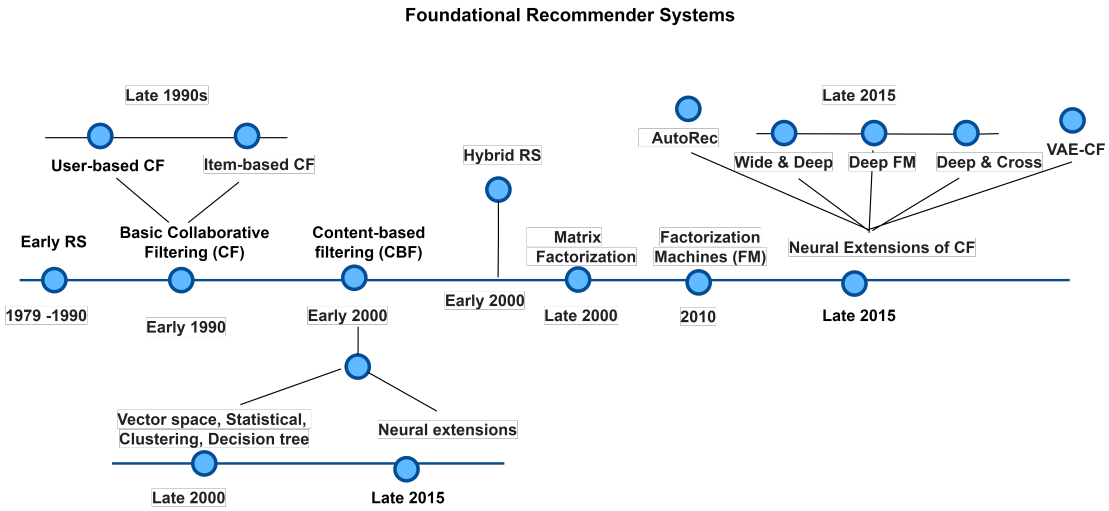
Alguns tipos de sistemas de recomendação se consolidaram como alicerces no estado da arte, ilustrados numa linha do tempo pela figura 11, sendo modelos mais antigos que estabeleceram as bases do estudo dos SR, citados por Raza et al. (2026):

- **Filtragem baseada em conteúdo (*Content-Based Filtering, CBF*):** uma estratégia de recomendação onde itens são sugeridos baseado na semelhança dos atributos do item com as preferências do usuário, que por sua vez é medida pela similaridade entre de vetores de características do item e os vetores de preferências do usuário. Tal medida se dá por técnicas que abrangem similaridade de cosseno, árvores de decisão e modelos neurais de *deep-learning*.

- **Filtragem Colaborativa (*Collaborative Filtering, CF*):** uma técnica cujo objetivo é de prever as preferências de um usuário baseado nas preferências de outros usuários semelhantes. Ela é subdivida em duas categorias que são os métodos de aplicação desse pretexto, sendo os métodos baseados em memória e em modelo. Do método baseado em memória, existem duas técnicas — filtragem de usuários e filtragem de itens — onde se calcula similaridades entre vetores de usuários ou de itens para efetuar a recomendação. Já no método baseado em modelo, são usadas técnicas como a fatoração de matrizes, onde a recomendação é dada por meio de uma aproximação entre vetores de características do usuário e do item.

- **Sistema Híbrido de Recomendação:** um sistema que combina diferentes técnicas de recomendação, como a filtragem baseada em conteúdo e filtragem colaborativa, com modelos de aprendizagem de máquina para recomendação, usando de técnicas como a soma ponderada de valores de recomendação dadas por funções de CF e CBF para obter uma recomendação final (RADFORD et al., 2018; ÇANO; MORISIO, 2017).

Figura 11 – Linha do tempo das técnicas de recomendação



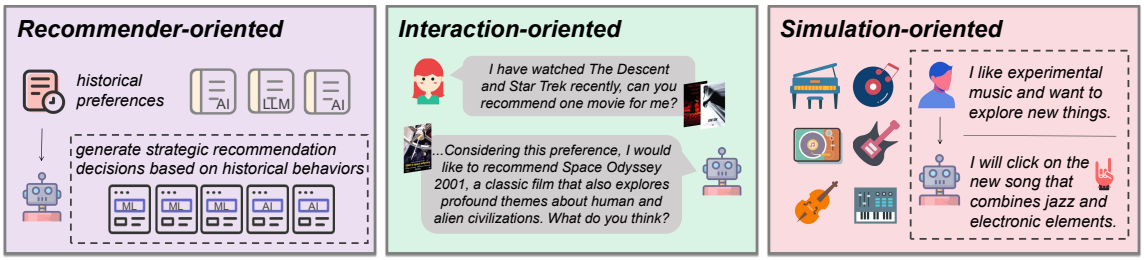
Fonte: Raza et al. (2026)

Na ultima década, a importância dos sistemas de recomendação tornou-se ainda mais destacada com a enormidade de dados produzida pelas interações de usuários online (*Big Data*), juntamente aos avanços produzidos no campo da Inteligência Artificial, cujas técnicas passaram a integrar mais a concepção de tais sistemas (RAZA et al., 2026). Peng et al. (2025) explica que, devido as capacidades de raciocínio e compreensão das relações de preferências de usuários, os LLMs foram adotados para contribuir no processo de recomendação, visto que seu método inerente de decisão é dado por nuances que vão além daquelas encontradas nas técnicas de pareamento de similaridades entre usuários e itens.

Desde então, vários métodos de incorporação de LLMs foram apresentados para produzir sistemas de recomendação, classificados por Peng et al. (2025) em três categorias (ilustradas na figura 12):

- **orientado como Recomendador:** a recomendação é produzida a partir de um planejamento que alavanca informações do histórico do comportamento do usuário, como contexto para uma geração direta pelo LLM. Tal contexto pode ser organizado e fornecido a LLM com ferramentas que vão tratar o LLM como o agente de recomendação direta.
- **orientado por Interação:** são métodos onde a recomendação é dada pelo LLM mediante interação conversacional, de forma que o modelo utilize suas habilidades de identificação das preferências do usuário para responder como em um diálogo, explicando o "raciocínio" na medida que gera a recomendação.
- **orientado por Simulação:** tratam-se de métodos cuja interação simula um papel de usuário que demonstra as suas preferências explicando o raciocínio motivador do usuário ter gostado ou não de um item, para que este raciocínio mantenha-se na memória contextual do LLM, a ser elicitado quando uma nova interação pede a inferência que diz se um item deve ser recomendado ou não, como ocorre no framework *AgentCF*(ZHANG et al., 2023).

Figura 12 – Métodos de recomendação com LLMs



Fonte: Peng et al. (2025)

3 Trabalhos Relacionados

Neste capítulo serão apresentados os trabalhos que se relacionam com a proposta introduzida por esta monografia. Será discutido sobre cada proposta de tais trabalhos, bem como sobre suas semelhanças na intenção de propor abordagens para a tarefa de recomendação utilizando do poder dos modelos de linguagem de grande escala e/ou combinando algoritmos encontrados no estado da arte da pesquisa de deep learning e de sistemas de recomendação.

3.1 Recomendação com modelo *Transformer* pré-LLMs

Num contexto imediatamente pré-explosão a notoriedade dos modelos de linguagem de grande escala, bem como da publicação de várias LLMs por diversos instituições como OpenAI e Google, Kang e McAuley (2018) apresentam uma proposta de sistema de recomendação de item seguinte, consistindo numa estrutura de rede neural artificial, nomeada SASRec, que integra uma versão customizada do mecanismo *self-attention*, originalmente apresentado na publicação do primeiro modelo de redes neurais *Transformers* que viriam a ser, por sua vez, base da constituição dos modelos de linguagem de grande escala. A estrutura segue como um modelo de rede neural para predição: no seu início, uma camada de *embeddings* para entrada das sequências de treino, que são os itens do histórico do usuário, representados ambos em valor de identificação, dentre um espaço vetorial que compreende todos os itens, e em valor posicional, sobre onde o item se localiza na sequência; para saída do modelo, define-se uma camada de predição de itens mais prováveis a serem escolhidos a seguir, pelo usuário alvo, que é a recomendação. O trunfo da proposta trata-se do uso do mecanismo *self-attention* para capturar o contexto de preferência do usuário a partir seus itens, ao computar paralelamente sua relevância quando presentes na sequência de histórico de interação, permitindo-se a identificação de características sobre a aparição de um dado item junto a outros itens específicos, também ao relacionar a relevância que um item pode ter, a depender da posição em que está localizado na sequência, visto que tal posição é definida pelo tempo mais recente ou antigo de quando foi consumido. Entre características apontadas como forças do modelo SASRec, em comparação com outras estratégias do estado da arte, como as redes neurais convolucionais e recorrentes, está o melhor tempo e eficiência de treinamento permitido pela execução paralela notável do mecanismo *self-attention* em GPUs, além da qualidade das características construídas no modelo por meio camadas *self-attention*, destacando-se o poder de identificação de similaridade entre itens de mesma categoria, de determina-

ção de relevância a itens a partir de sua recência, e da adaptabilidade em treinar com datasets sejam de espaço de itens denso ou esparso, devidamente ajustando seus pesos de atenção mediante tais características. As métricas avaliativas utilizadas são as mesmas encontradas nos trabalhos colocados para comparação, HitRate@10 e NDCG@10, as métricas utilizadas na avaliação de sistemas de recomendação, que aparecem nos trabalhos mencionados neste capítulo e que também foram adotadas na etapa de avaliação da proposta nesta monografia.

3.2 Recomendação com LLMs baseado na técnica *Zero-Shot*

Wang e Lim (2023) apresentam uma proposta de sistema de recomendação que utiliza uma estratégia zero-shot de prompt engineering para extrair a inferência de recomendação pela LLM GPT-3, sob demonstração com o dataset MovieLens 100K (HARPER; KONSTAN, 2015), conhecido dataset que consiste em análises de 1683 filmes feitas por um número de 943 usuários. Ele aponta como, apesar de ser promissora a estratégia de prompting Zero-Shot, extrair a inferência do modelo apenas apresentando itens candidatos para requisitar a recomendação, não produz resultados satisfatórios, visto que o modelo utilizado não é, especificamente, treinado com o objetivo de realizar a tarefa de recomendação. A ideia da proposta, então, é de utilizar as capacidades de inferência do modelo de linguagem de grande escala para extrair a definição tanto de contexto sobre os itens candidatos a serem recomendados, quanto sobre as preferências do usuário a partir de itens em seu histórico de consumo, para enfim utilizar estas definições numa prompt que requisita a recomendação pela LLM. Dado um usuário alvo, um conjunto de itens candidatos e um conjunto de itens do histórico do usuário alvo, esta estratégia se dá por um conjunto de requisições ao GPT-3, no modelo de inferência *text-davinci-003*, que são realizadas em três etapas sequenciais: cada etapa uma prompt, a servir para imbuir a prompt seguinte com contexto, onde a prompt construída na última etapa está estruturada com a requisição da inferência de recomendação para o usuário alvo. Ainda na composição desta estratégia, está o uso de técnicas de filtragem colaborativa (SARWAR et al., 2001; DWICAHYA; ROSA; NUGROHO, 2019), conhecidas como filtragem por usuário e filtragem por item, presentes como um módulo que antecede a requisição com as prompts. Estas técnicas tratam da definição do conjunto de itens candidatos a serem designados ao usuário alvo, a partir da filtragem dos itens do dataset, que neste caso são filmes assistidos. Na filtragem por usuário é computado a semelhança que todos usuários têm em comparação ao usuário alvo de recomendação, a partir dos vetores de itens consumidos, para então extrair um número m de usuários mais semelhantes e destes extrair um total de s itens mais consumidos para compor o conjunto de candidatos. De modo semelhante, na filtragem por itens é computada a semelhança que existe entre os itens do usuário

alvo e os itens de todos outros usuários do dataset, cada item sendo representado por um vetor de usuários que interagiram com dado item — definidos os itens mais semelhantes aos itens do usuário alvo, a partir deles são então extraídos um número *s* de itens mais populares, o que significa que são os mais interagidos, para compor o conjunto de candidatos. A tarefa de recomendação aqui trata-se da "recomendação de item seguinte", que consiste na predição do item mais provável que um usuário há de interagir a partir do seu histórico de interação, e na estratégia proposta, a prompt para recomendação requisita ao modelo *text-davinci-003* uma seleção de 10 itens que seriam da preferência do usuário, a partir do conjunto de itens candidatos. Ao comparar sua estratégia com abordagens para sistemas de recomendação propostos em outros trabalhos publicados, como o sistema SASRec, o autor mostra o benefício de incluir na prompt de requisição o contexto das preferências do usuário, bem como de integrar as técnicas de filtragem de itens para definir candidatos a recomendação, com resultados superiores mediante métricas avaliativas apresentadas nas outras propostas, nomeadamente, as métricas HitRate@10 e NDCG@10.

Liang et al. (2025) apontam os desafios que sistemas de recomendação podem ter em depender das capacidades de inferência das LLMs, nomeadamente a capacidade zero-shot para realizar a tarefa de recomendação, dado aspectos como: a limitação de inclusão de conjuntos de itens a depender de seu tamanho, visto que o tamanho do contexto a ser fornecido a LLM, via a prompt, é limitado e conjuntos muito grandes devem ser podados; a informação em texto dos itens recomendáveis, fornecida a prompt pode ser vaga ou não-estruturada, de forma a não representar o seu significado, ambiguidade que pode prejudicar a recomendação quando itens são informados diretamente por seus títulos, sem serem acompanhados de maior contexto; a geração não controlada de itens na recomendação, dado que o processo de geração de informação pelas LLMs ocorre de maneira não controlada sob dependência do seu contexto de pré-treinamento, então existe a possibilidade de que, itens de fora de um conjunto de recomendação, feito pelo sistema recomendador, sejam dados como recomendação. É apresentada então uma proposta que busca lidar com tais desafios, ainda aproveitando do poder de inferência zero-shot dos modelos de linguagem de grande escala, para produzir um sistema de recomendação zero-shot. No intuito de estruturar o contexto de itens recomendáveis, é proposto o framework batizado TaxRec, que extrai, via LLM, a taxonomia dos itens que serão contexto para a recomendação, a fim de produzir a categorização e agrupamento destes itens, e isto ocorre na execução de duas etapas. Na primeira etapa, nomeada *One-time Taxonomy Categorization,* por meio de uma prompt à LLM são definidas características sobre o contexto do dataset escolhido para o sistema de recomendação, seja de livros, filmes, ou outros e, a partir destas características, com uma nova prompt a LLM produz a categorização dos itens do dataset. Com os itens estruturados em categorizações como gênero, tema e

idioma, na segunda etapa essas informações são usadas numa prompt que extrai a recomendação pela LLM, formatada com a taxonomia definida, dessa forma dando contexto tanto sobre o significado dos itens de preferência do usuário alvo, quanto sobre os itens candidatos de recomendação. A prompt de recomendação requisita uma lista com um número de *k* itens para o usuário alvo, e esta saída pode ser analisada sob um ranking *top-k* de itens relevantes, que serão, desta forma, recomendados ao usuário. O TaxRec foi então avaliado por conhecidas métricas do estado da arte do sistemas de recomendação (PENG et al., 2025), Recall@K e NDCG@K, com valores K de 1, 5 e 10, também com variação das LLM escolhidas, sendo o modelo Llama 2 e o modelo GPT-4. Analisando a performance sob um dataset do contexto de filmes, MovieLens 100K, e um dataset de livros, BookCrossing, mostrou-se a performance ao comparar com baselines de recomendação zero-shot tanto compostos por LLMs ou não, conseguindo resultados melhores diante de propostas que não incluíam na prompt zero-shot as informações de taxonomia introduzidas pelo TaxRec. Ao comparar com um sistema chamado DirectRec, uma variação do TexRec que gera a recomendação informando à LLM os itens candidatos e de histórico do usuário sem as estruturações de taxonomia propostas, foi obtido Recall@10 de 0.180 e NDCG@10 de 0.099, contra Recall@10 de 0.300 e NDCG@10 de 0.157 no TexRec.

3.3 Ajuste Fino de LLMs para Recomendação

O sistema de recomendação apresentado por Yue et al. (2023), nomeado LlamaRec, é constituído pelo modelo *Linear Recurrent Units for Recommendation* (LRURec) (YUE et al., 2024), produzido para a tarefa de recomendação sequencial de itens, em combinação com o modelo de linguagem de grande escala Llama 2. Na estrutura do LlamaRec, dado um conjunto de itens de um usuário, seu histórico de interação e um conjunto de todos itens recomendáveis, são colocados dois estágios sequenciais para realização da recomendação: primeiro com um *Retriever*, ou recolhedor, papel cumprido pelo modelo de recomendação sequencial LRURec, que passa por um treinamento autoregressivo de predição de itens, e é usado para selecionar um conjunto de itens candidatos mais prováveis para escolha pelo usuário alvo; segundo e último, com um *Ranker*, ranqueador representado pelo LLM Llama 2, que passa por treinamento de *instruction-tuning* com as instruções, isto é, prompts estruturadas, feitas para que o modelo realize o ranqueamento dos itens candidatos fornecidos junto ao histórico do usuário. Desta forma, a saída deste ranqueamento representa então a recomendação realizada pelo framework. A partir deste sistema, Choi e Kim (2025) apresentam melhorias ao desempenho da recomendação feita pelo LlamaRec, propondo substituir o componente *Ranker* sendo Llama 2, para o uso do modelo Llama 3 sob a mesma tarefa de *fine-tuning*, seguindo o treinamento do modelo com as instruções usadas na

tarefa ranqueamento do LlamaRec. Para avaliação deste trabalho de melhoria, foram utilizadas métricas comumente encontradas em trabalhos que apresentam sistemas de recomendação e que também estavam presentes na avaliação publicada do LlamaRec original, sendo elas *Mean Reciprocal Rank* (MRR@k), *Normalized Discounted Cumulative Gain* (NDCG@k) e *Recall* (R@k), onde k representa o valor do alcance Top-K ao avaliar os itens recomendados, aqui usado Top-5 e Top-10. Tal avaliação foi feita ao analisar dois aspectos, um sendo a performance de recomendação do *Ranker*, onde se avalia apenas as recomendações feitas a partir das seleções dadas pelo *Retriver* que continham os itens de predição correta, e o outro sendo a performance de recomendação geral, onde se avalia as recomendações pelo *Ranker* feitas a partir de ambas seleções do *Retriever* que tiveram itens de predição correta e seleções de itens cujos rótulos não eram corretos. Foram demonstrados ganhos de performance em ambas análises quando comparados ao LlamaRec usando Llama 2 contra Llama 3 como ranqueador de itens para recomendação, onde, em avaliação de performance do *Ranker* sob uma versão do dataset MovieLens 100K, consegue-se valor MRR@10 de 0.3235, NDCG@10 de 0.4142 e Recall@10 de 0.7142 contra o sistema com Llama 2, que consegue MRR@10 de 0.2083, NDCG@10 de 0.3184 e Recall@10 de 0.6825. Esses ganhos demonstraram a hipótese de que o uso de um LLM melhorado no sistema de recomendação pode consequentemente dar melhores resultados de recomendação.

Ebrat et al. (2025) vai além, ao introduzir uma proposta de sistema de recomendação que visa unificar um conjunto de técnicas de recomendação, fazendo proveito do poder de contextualização das LLMs nas múltiplas etapas da estrutura deste sistema híbrido, que consiste em dois estágios. Com estrutura demonstrada a partir de experimentos com o dataset MovieLens 100K, no estágio de pré-processamento, os dados não-tratados do dataset são transformados em informações de caracterização dos itens, extraídas do LLM com input de metadados textuais que informam aspectos como título e gênero, bem como de caracterização das preferências dos usuários, também extraídas do LLM em prompts sequenciais que requisitam inferências sobre as preferências dos usuários, dado os itens de seu histórico. Com as caracterizações definidas sobre os itens e sobre os usuários em relação aos itens, são produzidos esquemas textuais para detalhamento do que foi extraído, numa estrutura *Markdown* com as seções: *Overview*, que inclui um resumo das preferências do usuário ou da narrativa do item; *Attributes*, que detalha caracterizações como gêneros do item ou gêneros que são da preferência do usuário; *Description*, onde são detalhadas possíveis preferências temáticas que o usuário tem, ou que são da narrativa do item; e *Dislikes*, onde se indica atributos não desejáveis pelo usuário (usuário que gosta de filmes humor e aventura e desgosta dos gêneros terror, suspense, por exemplo) ou atributos de categoria ausentes no item (um item que representa um filme infantil de comédia, pode

não possuir atributos de terror). Estes esquemas textuais de detalhamento dos itens e usuários são transformados em representações de *embeddings*, para serem usados na geração de um grafo bipartido usuário-item, que relaciona as duas entidades, os vértices representando usuários ou itens, e as arestas representado a avaliação positiva ou negativa dada pelo usuário a um item. Este grafo serve como *embedding* de input para o treinamento de um modelo de rede neural de grafos GAT, ou *Graph Attention Network*, constituída de três camadas formadas para a tarefa de filtragem colaborativa, compostas de sub-camadas *attention* que computam, de forma bidirecional, a relação entre cada par de vértices usuário-item. O modelo é otimizado sob uma função de perda consistindo na operação de filtragem colaborativa *Bayesian Personalized Ranking* (RENDLE et al., 2012), que ranqueia em relevância os itens para os usuários, combinado com um regularizador que realiza a operação de similaridade em cosseno, para aproximar os vértices que representam pares usuário-item cujas arestas são de avaliação positiva. Após a realização da filtragem colaborativa por meio do modelo GAT, no estágio de pós-processamento é definido um conjunto de operações com o LLM, variantes de execução independente, cujo objetivo é de refinar e re-ranquear as seleções de itens candidatos, para se obter os conjuntos finais de itens que serão a recomendação. Tais variantes incluem: a construção de uma prompt a ser fornecida ao LLM e formatada com o contexto das preferências do usuário sobre o conteúdo dos itens em gênero e temática, junto a descrições dos itens candidatos; o uso de uma árvore de busca binária onde pares de itens são comparados pelo LLM sob o contexto das preferências do usuário; a designação de pontuação de relevância de cada item candidato por avaliação do LLM, realizada em lotes de itens candidatos, onde as pontuações finais são normalizadas a partir da média de seu respectivo lote, para então definir-se a recomendação com os itens de maior relevância. Apesar de não citar qual modelo foi especificamente usado para obter os resultados mostrados na publicação, foi apontado que o modelo OpenAi 4.1-mini superou em performance os outros modelos utilizados, Gemma-3 4B e o4-mini, citando performance comparável entre esses dois, destacando a eficiência do Gemma-3 diante do o4-mini que possui tamanho maior. Os experimentos realizados mostraram ganhos de performance ao analisar os resultados deste sistema, com as métricas *Precision*, *Recall*, *NDCG* e *MAP,* comparados a performance dos baselines que são modelos de redes neurais de grafos para recomendação, como o NGCF(WANG et al., 2019) e o LightGCN (HE et al., 2020), obtendo, na análise de performance onde os usuários têm menos de 10 interações, Recall@10 de 0.126 e NDCG@10 de 0.253 para o NGCF, contra Recall@10 de 0.183 e NDCG@10 de 0.335 para o sistema proposto. Foram apontados como causas da performance superior, o uso das caracterizações sobre itens e usuários gerados por LLM, que oferecem rico contexto semântico para a recomendação, bem como o uso da representação textual de itens e usuários no esquema estrutural proposto para ini-

cialização do treinamento do modelo GAT construído.

3.4 Semelhanças e Diferenças com o Sistema Proposto

Os trabalhos apresentados trazem propostas originais que buscaram expandir os limites dos sistemas de recomendação, e nos casos onde a pesquisa dos LLMs já se difunde, buscou-se incluir nas estruturas de seus sistemas as capacidades dos modelos de linguagem de grande escala. Com exceção do trabalho de Kang e McAuley (2018), todos eles almejar utilizar das capacidades de inferência *zero-shot* dos LLMs. Mesmo que separem, sob técnincas diferentes, as etapas de extração e/ou definição do contexto que dá o teor de personalização ao usuário-alvo, tais arquiteturas convergem ao objetivar o fornecimento desse contexto ao LLM para que gere a recomendação, aplicando a técnica de engenharia de prompt *zero-shot*.

O sistema proposto por este trabalho, é dado por uma arquitetura que aplica técnicas de engenharia de prompt diferentes da *zero-shot* no passo de geração de recomendação pelo LLM. A arquitetura em questão é baseada naquela apresentada por Wang e Lim (2023), de onde se reutiliza o componente idealizado de filtragem colaborativa, que define o contexto de preferências do usuário-alvo a partir, primariamente, de itens candidatos escolhidos de acordo similaridade aos interesses do usuário-alvo. A partir desta definição de contexto, aplica-se técnicas as *one-shot* e *chain-of-thoughts* distintas ao que se encontra nos trabalhos relacionados, além de se introduzir estratégias *zero-shot* criadas aqui, propostas para a lógica comparativa deste trabalho.

3.5 Conclusão

A partir dos trabalhos discutidos neste capítulo, é possível investigar sobre as diversas formas em que são usadas e aprimoradas as técnicas difundidas no estado da arte de sistemas de recomendação, como na filtragem colaborativa, na extração de preferências do usuário via LLMs, no uso de grafos bipartidos para relacionar usuários a itens e na exploração do potencial existente na técnica zero-shot na engenharia de prompts. Compreende-se também sobre o exercício de demonstração comparativa entre diferentes propostas, constatando-se as métricas recorrentes usadas entre todos os experimentos.

Conclui-se, portanto, que a tendência dos sistemas de recomendações nesta era de LLMs, é de aproveitar o poder de compreensão de linguagem natural de tais modelos, aplicando essa capacidade a partir de técnicas diferentes. Nota-se que há uma busca continuada da aplicação do ajuste fino dos LLMs com treinamento sob medida para as intruções esperadas, como maneira de melhor aproveitar a capacidade

zero-shot com integração de contexto pré-produzido.

Tabela 1 –
Resumo dos trabalhos relacionados

Trabalho	**Modelo generativo utilizado**	**Técnica aplicada**
Kang e McAuley (2018)	-	redes *Transformers*, *self-attention*
Wang e Lim (2023)	GPT-3	Filtragem colaborativa, zero-shot
Choi e Kim (2025)	Llama 3	Ajuste fino, zero-shot
Liang et al. (2025)	GPT-4, Llama 2	Extração de taxonomia de itens, zero-shot
Ebrat et al. (2025)	4.1-mini, 04-mini, Gemma-3 4B	rede *attention* de grafos, grafos bipartidos, árvore de busca binária, zero-shot
Este trabalho	**gpt-3.5-turbo-instruct**, **LLama 3.1**, **Gemma 3**	**Filtragem Colaborativa**, **zero-shot**, **one-shot**

Fonte: o Autor

4 Proposta

A proposta deste trabalho é de produzir uma análise comparativa entre diferentes técnicas de prompt engineering sendo aplicadas no framework do sistema de recomendação apresentado por Wang e Lim (2023), que utiliza as técnicas de filtragem colaborativa e engenharia de prompt *zero-shot*, para extrair o potêncial dos LLMs como recomendadores. O objetivo é de propor estratégias de *prompt engineering* que possam substitir o módulo de estruturação de prompt apresentado na proposta original, que será o ***baseline*** de comparação, para investigar o potencial de melhorias de performance quando se utilizam os LLMs como recomendadores. Este *framework* foi escolhido pois sua abordagem sujeita-se fortemente a capacidade bruta que o LLM tem de inferir a recomendação, visto que os itens são recomendados mediante solicitação direta via prompt, o que dá abertura para explorar o potencial de melhoria com técnicas de engenharia de prompt diferentes.

Segue-se então a lógica arquitetural dada por Wang e Lim (2023), para reaproveitar o framework desse sistema de recomendação. Como discutido no capítulo anterior, a proposta deste trabalho combina as técnicas de filtragem colaborativa — filtragem por usuários e filtragem por itens — com uma estratégia zero-shot de engenharia de prompts, chamada *three-step prompting*, para realização da tarefa de recomendação. Com a filtragem colaborativa define-se os itens candidatos a partir do conceito de semelhança entre os usuários ou entre os itens da base de dados utilizada, sendo feita na filtragem por usuários a partir de co-ocorrências de itens em seus históricos de interação e, na filtragem por itens, sendo feita a partir de co-ocorrências de interações pelos usuários. Estas duas técnicas são definidas como variantes independentes a serem aplicadas na execução do framework, entretanto, para este estudo comparativo, foi escolhido utilizar apenas a filtragem por usuário como forma de restringir o escopo da avaliação.

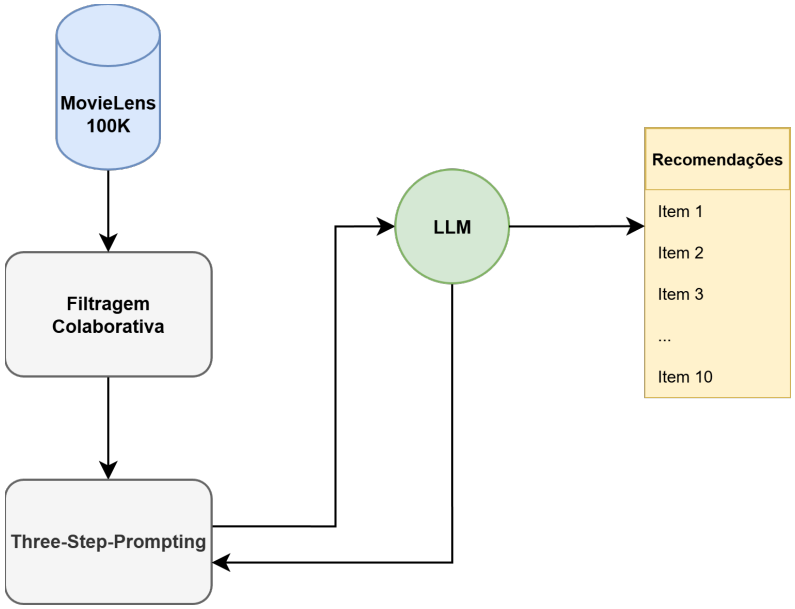
Figura 13 ilustra a arquitetura do sistema original. Ela consiste de um componente para seleção de itens candidatos com filtragem colaborativa, e outro que interage com o LLM para construção da prompt de recomendação e geração da recomendação:

- **Componente de Filtragem Colaborativa:** responsável pela criação das amostras de itens do histórico de interação do usuário alvo e seus itens candidatos. Aqui se aplica as operações de filtragem colaborativa, sendo nas variantes de filtragem por usuário ou filtragem por itens. A base de dados MovieLens 100K, utilizada originalmente, trata-se de um conjunto de registros de análises de filmes feitas por 943 usuários, entre 1682 filmes. Cada registro se refere, efetivamente,

à interação que um usuário teve, representando assim o conjunto de *feedback* implícito dos usuários. Desta forma, cada *feedback* implícito é definido simplesmente pela existência da relação de interação entre um usuário e o item avaliado. A partir das operações de filtragem colaborativa, explicada em detalhes no capítulo de Experimentos, as amostras são construídas ao se definir, para cada usuário do *dataset*, o conjunto dos itens interagidos e o conjunto de itens candidatos.

- **Componente *Three-step-prompting***: onde se executa a estratégia *Three-step-prompting*. Executando três prompts, forma-se contexto das preferências do usuário e gera-se a recomendação. Isso ocorre numa maneira sequencial, onde nas duas primeiras prompts se extrai inferências de personalização do usuário-alvo de recomendação, para serem usadas na ultima prompt de recomendação.

Figura 13 – Arquitetura do sistema de recomendação de Wang e Lim (2023)



Fonte: o Autor

A estratégia *three-step prompting*, consiste na construção sequencial de uma prompt que será fornecida para que o LLM realize a inferência de recomendação, possuindo contexto de preferências do usuário alvo, os itens do seu histórico e itens candidatos. No contexto da base de dados de análise de filmes MovieLens 100K, o processo de contrução da prompt de recomendação é feita em três etapas de requisição ao LLM, portanto com três prompts diferentes, da seguinte forma:

1. Constrói-se a primeira prompt com os itens candidatos para recomendação, junto ao histórico de interação do usuário, para requisitar ao LLM a inferência das pre-

ferências do usuário alvo, perguntando quais as características são mais relevantes dado os itens do seu histórico de interação;

2. A segunda prompt repete toda a estrutura da prompt anterior, concatenando a resposta da primeira inferência, para requisitar uma seleção de cinco itens relevantes dentre os itens do histórico de interação, dado as preferências obtidas, ordenados por maior preferência;

3. A terceira prompt é constituída de toda a estrutura da prompt anterior, também concatenando a resposta da segunda prompt, para, finalmente, requisitar a recomendação de dez itens, entre os itens candidatos, similares aos itens da seleção de itens relevantes, num formato que cada item recomendado seja respectivo a um item da seleção.

Snippet 4.1 – pseudocódigo da estratégia *three-step-prompting*

```
FUNCAO RECOMENDACAO_THREE_STEP_PROMPTING(itens_historico,
  itens_candidados, template_1, template_2, template_3)

   prompt_1 <- PREENCHER_TEMPLATE(itens_historico, itens_candidatos)
   preferencias_usuario_alvo <- REQUISITAR_LLM(prompt_1)

   prompt_2 <- PREENCHER_TEMPLATE(itens_historico, itens_candidatos,
   preferencias_usuario_alvo)
   itens_historico_preferidos <- REQUISITAR_LLM(prompt_2)

   prompt_3 <- PREENCHER_TEMPLATE(itens_historico, itens_candidatos,
   preferencias_usuario_alvo, itens_historico_preferidos)
   recomendacao <- REQUISITAR_LLM(prompt_3)

   RETORNAR recomendacao
```

O resultado deste processo mostrado com o pseudocódigo 4.1, é uma prompt que mantém todo o histórico de inferências feitas pela LLM e que contém o contexto das preferências do usuário, dado pela identificação de características relevantes e a demonstração de itens representativos de tais características. Esta ultima prompt é, enfim, utilizada para produzir a recomendação. A estratégia *three-step-prompting* é um objeto de comparação deste trabalho, sendo substituída por variantes de estratégias criadas aqui, dadas por prompts feitas para representar tanto variações de exploração do potencial das capacidades zero-shot de engenharia de prompt, quanto para representar a aplicação de outras técnicas que, por sua vez, poderiam melhorar a performance de recomendação por modelos de linguagem de grande escala, buscando, em paralelo, oferecer alternativas mais simplificadas para realização desta tarefa.

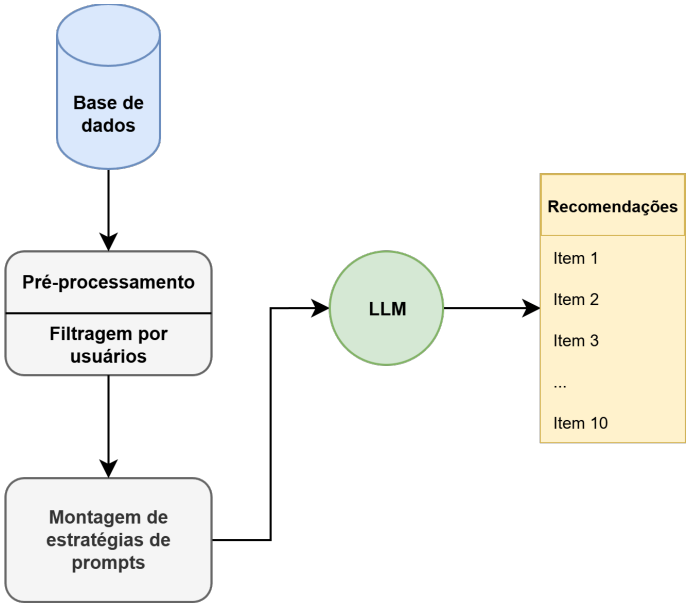
Dado o framework original, em primeira instância serão apresentadas quatro estruturas de prompts usadas para explorar a performance de recomendação de LLMs, quando realizam inferências a partir das diferentes técnicas de engenharia de prompt, representadas pelas respectivas estruturas construídas. Nesta instância é proposta a análise do impacto de performance, derivado da substituição da estratégia de engenharia de prompt original, pelas estratégias de prompts originadas por este trabalho, entretanto, realizando experimentos numa base de dados diferente da utilizada no *baseline* — sendo em dois dos subconjuntos que compõem a base de dados Amazon Reviews Dataset (NI; LI; MCAULEY, 2019), nomeados Beauty e Luxury Beauty. Tal análise é mais especificamente sobre a tarefa de recomendação de item seguinte (SRILAKSHMI; CHOWDHURY; SARKAR, 2022; GRBOVIC et al., 2015), abordada pelo *baseline*, onde se recomenda o que seria o próximo item de mais provável interação por um usuário alvo. Os experimentos realizados buscam aplicar as métricas avaliativas apresentadas no *baseline*, para a comparação entre os resultados de performance obtidos com o uso de cada estratégia de prompt aplicada neste sistema de recomendação, sob análise, nesta etapa, na base de dados provida por Ni, Li e McAuley (2019).

A partir das descobertas obtidas com os resultados dos experimentos na base de análises de itens da amazon, busca-se definir a escolha de uma das estratégias de prompt criadas, para que, então, esta seja aplicada numa nova análise comparativa de performance de recomendação, desta vez sendo feita em cima da base de dados MovieLens 100K (HARPER; KONSTAN, 2015), dataset por onde se avaliou originalmente o framework de recomendação do *baseline*. Com estes experimentos comparativos, busca-se constatar as possíveis melhorias ou pioras de performance obtidas com as estruturas de prompts apresentadas, além de análisar quais aspectos as prompts envolvidas possuem como causalidade de impacto no desempenho de recomendações feitas por LLMs.

4.1 Modificações no *framework* do sistema de recomendação

Para este trabalho, propõe-se a modificação do componente da estratégia de engenharia de prompt, de modo que se aplique as quatro estratégias de engenharia de prompt como variantes para realização da tarefa de recomendação. Também se utiliza uma base de dados externa para produção de análise comparativa, sendo a base de dados Amazon Reviews Dataset.

Figura 14 – Arquitetura do sistema de recomendação



Fonte: o Autor

Figura 14 ilustra a arquitetura do sistema modificada para uso nos experimentos de recomendação, de forma que se substitua a estratégia de prompt dada pelo *baseline*. Ela consiste de dois componentes que antecedem a chamada a API do LLM:

- **Componente de pré-processamento/filtragem por usuários:** mantém-se como na arquitetura original, entretanto aqui se engloba a operação de pré-processamento dos dados brutos da base de dados, transformados em dados passíveis de consulta, de forma que seja possível selecionar usuários do *dataset* e os seus itens. Ambas bases de dados em questão tratam-se de coleções de análises de itens — uma sendo para filmes e, a outra para produtos — com registros sendo estruturados por atributos como *id* do usuário, *id* do item e data de avaliação, por onde se computa o *implicit feedback* dos usuários. Realizado o pré-processamento, as amostras são construídas assim como no *framework* original e, aqui se utiliza a implementação da filtragem por usuários feita por Wang e Lim (2023), para definir itens candidatos para cada usuário-alvo de recomendação.

- **Componente de montagem de estratégias de prompts**: esse é o componente responsável pela construção das prompts definidas de cada estratégia a ser usada para a recomendação, tendo também a responsabilidade de realizar as requisições à **API** do LLM, fazendo envio das prompts e o recebimento das respectivas respostas. Para cada estratégia de prompts, são definidos templates a serem processados nesta etapa, onde se insere os conjuntos de itens interagidos e itens candidatos do usuário alvo, definidos em cada amostra pelo componente anterior, além de inclusões de respostas do LLM quando lidando com estraté-

gias de etapas de prompts sequenciais. Este componente recebe, portanto, cada resposta final requisitada via estratégias de prompt e, dessa maneira, recebe a resposta de recomendação pelo LLM.

É, portanto, reaproveitado o componente de filtragem por usuários do framework original, dado pelo baseline, para definir a seleção de itens candidatos que, então, são utilizados para recomendação com as estratégias de prompt propostas por este trabalho, num processo mostrado com o pseudocódigo 4.2. Com esta modificação, pretende-se investigar o impacto que as novas propostas de estratégias de prompt vão ter na tarefa de recomendação, para avaliar se é possível conseguir melhoria no desempenho de recomendação com estas estratégias menos custosas e de aplicação de técnicas de engenharia de prompt distintas.

Snippet 4.2 – Pseudocódigo do sistema de recomendação avaliado

```
FUNCAO RECOMENDAR_COM_ESTRATEGIA_PROMPT(estrategia_prompt,
  dados_itens_usuarios)

   colecao_pre_processada <- PRE_PROCESSAR_DADOS(
  dados_itens_usuarios)

   lista_amostras_itens_usuarios <- FILTRAGEM_POR_USUARIOS(
  colecao_pre_processada)

   template_prompts <- CARREGAR_TEMPLATE_ESTRATEGIA(
  estrategia_prompt)

   PARA CADA amostra_usuario_alvo EM lista_amostras_itens_usuarios

       prompt_formatada <- FORMATAR_PROMPT(amostra_usuario_alvo)

       recomendacao <- REQUISITAR_LLM(prompt_formatada)
```

4.2 Técnicas de engenharia de prompt utilizadas

Foram criadas quatro estratégias de prompts para substituir a *three-step prompting*, discutida no capítulo anterior. Duas destas estratégias representam uma variação da técnica zero-shot, uma representa a aplicação da técnica one-shot e a última representa a técnica chain-of-thoughts. Estas novas estratégias ainda seguem o princípio dado pela estratégia do baseline, de incluir o conjunto de itens do histórico de interação

e o conjunto de itens candidato para realizar a recomendação. Entretanto, elas diferem na maneira em que vai se incluir, ou não, o contexto de preferências do usuário.

4.2.1 A estratégia do baseline: *Three-step prompting*

Esta é a estratégia proposta por Wang e Lim (2023), baseline da comparação de performance com as estratégias propostas por esta análise comparativa. Ela consiste nos três seguintes templates, a serem usados de forma sequencial:

```
Candidate Set (candidate products of <product category> category): <candidate set>.
The products I have reviewed (reviewed of <product category> category) (Format: [Product - Summary]): <reviewed set>.
Step 1: What features are most important to me when selecting a product (Summarize my preferences briefly)?
Answer:
```

```
Candidate Set (candidate products of <product category> category): <candidate set>.
The products I have reviewed (reviewed of <product category> category): <reviewed set>.
Step 1: What features are most important to me when selecting a product (Summarize my preferences briefly)?
Answer: <prompt 1 inference>.
Step 2: Selecting the most featured products from the reviewed products according to my preferences (Format: [no. a reviewed product.]).
Answer:
```

```
Candidate Set (candidate products of <product category> category): <candidate set>.
The products I have reviewed (reviewed of <product category> category): <reviewed set>.
Step 1: What features are most important to me when selecting a product (Summarize my preferences briefly)?
Answer: <prompt 1 inference>.
Step 2: Selecting the most featured products from the reviewed products according to my preferences (Format: [no. a reviewed product.]).
Answer: <prompt 2 inference>.
Step 3: Can you recommend 10 products from the Candidate Set similar to the selected products I've reviewed (Format: [no. a reviewed product - a candidate product])?.
Answer:
```

4.2.2 *Two-step prompting*

Numa lógica semelhante a estratégia original de três passos de prompts, foi construída uma estratégia zero-shot, que consiste em dois passos de requisição ao LLM, porém sem a inclusão da descrição dos passos na estrutura das prompts, também não existindo um passo que extraia diretamente uma resposta sobre as preferências do usuário. É dada, portanto, pelos seguintes templates:

```
Candidate Set (candidate products of <product category> category): <candidate set>.
The products I have reviewed (reviewed of <product category> category): <reviewed set>.
Given the products I have reviewed
Select the most relevant products according to what my preferences would be (Format: [no. a review product.], ordered from
highest to lowest relevance).
Answer:
```

```
Candidate Set (candidate products of <product category> category): <candidate set>.
The products I have reviewed (reviewed of <product category> category): <reviewed set>.
Given my Candidate Set
Given my most relevant products: <previous inference>
Recommend 10 products from the Candidate Set similar to my most relevant products (Format [no. a relevant product - a candidate product])
Answer:
```

4.2.3 *Single-step prompting*

Esta estratégia zero-shot consiste no uso de uma única prompt, que busca elicitar do LLM uma lógica implícita sobre a necessidade de inferir o contexto das preferências do usuário alvo, enviesando, além disso, sobre a necessidade de que a recomendação seja feita a partir da inferência de 10 itens mais relevantes, dado histórico de interação do usuário alvo. Desta maneira, buscou-se extrair do LLM uma reprodução lógica semelhante aos mesmos passos de extração de contexto reproduzidos no baseline, a partir de uma prompt que tenta condensar aquelas etapas. É dada pelo seguinte template:

```
Candidate Set (candidate products of <product category> category): <candidate set>.
The products I have reviewed (reviewed of <product category> category): <reviewed set>.
Recommend 10 products from the Candidate Set similar to 10 of the most relevant products for me from the products I have reviewed (Format: [no. a relevant product - a candidate product])
Answer:
```

4.2.4 *Chain-of-thoughts prompting*

A construção desta estratégia, foi baseada na aplicação de uma técnica *chain-of-thoughts* de engenharia de prompt apresentada por Kojima et al. (2023). Na única prompt que constitui essa estratégia, a tarefa de recomendação a ser executada é dada com uma pergunta, estruturada numa declaração textual de problemática, que apresenta os conjuntos de itens de interação do usuário alvo, junto aos itens candidatos e que deve ser resolvida pelo LLM. Descrita a problemática, ao final da prompt, após a descrição da pergunta, se insere uma frase chave: "*Let's think step-by-step*". Esta frase implica ao LLM a necessidade de descrever a lógica de etapas tomadas para influenciar a decisão do itens recomendados. A prompt se dispõe pelo template a seguir:

```
The user chooses products in the <product category> category to consume based on its preferences according to previously reviewed products.
The user has a set of previously reviewed products
(
<reviewed set>
)
and a set of candidate products
(
<candidate set>
).
What would be a selection of 10 recommended product for the user, based on its preferences, following the format [no. a relevant product - a candidate product]?
Let's think step by step.
```

4.2.5 *One-shot prompting*

Na única prompt que compõe a estratégia criada, é inserido um exemplo de execução da tarefa de recomendação almejada, seguindo uma premissa de forma semelhante às outras estratégias criadas, com a inclusão de conjunto de itens do histórico de interação do usuário alvo e os itens candidatos a recomendação. O exemplo é estruturado para representar a pergunta que solicita a recomendação, seguido da resposta

esperada. Em sequência à inserção do exemplo, incluí-se o que seria um novo exemplo de pedido de recomendação, só que a resposta, por sua vez, é deixada como a conclusão lógica a ser inferida pelo LLM. O template desta prompt é estruturado da seguinte forma:

```
A user with candidate set of movies
(
Under Siege 2: Dark Territory
Natural Born Killers
The Three Musketeers
Batman Returns
GoldenEye
Batman Forever
Days of Thunder
Star Trek III: The Search for Spock
Full Metal Jacket
Star Trek VI: The Undiscovered Country
The Crow
Speed
Money Train
Stargate
True Lies
Conan the Barbarian
Under Siege
First Knight
Batman
)
and a set of previously reviewed movies
(
The Maltese Falcon
Con Air
Romy and Michele's High School Reunion
Anastasia
Grosse Pointe Blank
The Fifth Element
Starship Troopers
Wild America.
)
receives as recommendation the following 10 movies from the candidate set of movies, based on his preferences on his previosly reviewed movies, formatted as [no. a relevant movie - a candidate movie]:
(
1. The Fifth Element - Star Trek III: The Search for Spock
2. Starship Troopers - Star Trek VI: The Undiscovered Country
3. Con Air - Speed
4. Grosse Pointe Blank - True Lies
5. Wild America - Days of Thunder
6. The Fifth Element - Batman Returns
7. Starship Troopers - GoldenEye
8. Con Air - Under Siege
9. Grosse Pointe Blank - The Crow
10. Wild America - First Knight
)

Another user with candidate set of movies
(
<candidate set>
)
and a set of previously reviewed movies
(
<reviewed set>
)
receives as recommendation the following 10 movies from the candidate set of movies, based on his preferences on his previosly reviewed movies, formatted as [no. a relevant movie - a candidate movie]:
```

Admite-se que a aplicação da estratégia *One-shot*, usada nas duas bases de dados, envolveu um ajuste da prompt para o contexto do dataset MovieLens 100K, de forma que a prompt se refere aos itens como filmes, algo observável ao analisar os exemplos de prompts formatadas A.7 e A.6 no apêndice A. Para tornar mais genérica a aplicação dessa e das outras estratégias, poderia ter-se seguido a estruturação de prompt feita para o dataset de análises Amazon, que se refere a qualquer item como um produto de uma devida categoria — no exemplo do dataset de filmes, poderia se descrever cada item como sendo um produto da categoria "*movies*". Essa estruturação tornaria generalizada a aplicação das estratégias em qualquer base de dados de qualquer contexto de itens a serem recomendados, bastaria incluir a categorização do tipo

de produto recomendável que o item se refere, seja música, livros, ou classes gerais de produtos.

4.3 Modelos LLM escolhidos

Os modelos escolhidos para esta análise comparativa são da categoria dos LLMs *instruction-tuned*, sendo variantes nos experimentos de recomendação. Tratam-se de modelos que passam por uma operação específica de treinamento, denominada *instruction-tuning*. Esta operação se dá pelo treinamento de ajuste da responsividade que o LLM tem ao lidar com instruções fornecidas pela prompt, para reforçar esse viés de compreensão do modelo (JIANG et al., 2024; ZHANG et al., 2026). Foram utilizados os modelos **GPT-3.5-Turbo-Instruct**, da OpenAI, **Llama 3.1 8B Instruct**, do Meta e o modelo **Gemma-3-4B-IT** da Google, versões dos modelos "principais" respectivos, GPT-3, Llama 3 e Gemma 3. Eles variam em tamanho de parâmetros e autoria organizacional, escolhidos de forma a oferecer um balanço comparativo das capacidades fornecidas pelas empresas que os produziram.

O modelo de inferência *text-davinci-003*, variante do GPT-3 usado pelo baseline, foi depreciado no início do ano de 2024 pela organização OpenAI, que indicou o modelo *gpt-3.5-turbo-instruct* como substituto direto (OPENAI, 2024). Este modelo foi escolhido para seguir a intuição dada pelo baseline, de utilizar o modelo *text-davinci-003* para a recomendação. Com exceção do *gpt-3.5-turbo-instruct*, de uso exclusivamente via API de autoria própria do OpenAI, os modelos são abertos, com quantidade de parâmetros reduzido em comparação às suas contrapartes providas de números razoavelmente maiores — com 8 bilhões de parâmetros na versão Llama 3.1 8B Instruct diante dos 405 bilhões de parâmetros da versão Llama 3.1 405B Instruct (GRATTAFIORI et al., 2024) e com 4 bilhões de parâmetros na versão Gemma-3-4B-IT, em comparação aos 27 bilhões na versão Gemma-3-27B-IT (TEAM et al., 2025). Essa escolha foi motivada pela decisão de executar estes modelos em ambiente local e, por permitirem uma análise sobre a performance que poderia se obter a partir de modelos com escala menor de parâmetros.

4.4 Conclusão

Neste capítulo foi apresentada a proposta da análise comparativa realizada neste trabalho, onde se reaproveita do sistema de recomendação apresentado originalmente por Wang e Lim (2023), substituindo o aspecto de engenharia de prompt da arquitetura original, para aplicar novas propostas de engenharia de prompt, almejando melhorias de desempenho da tarefa de recomendação, a partir de um estudo que analise o impacto que cada estratégia de prompt tem ao ser empregada. Apresentou-se

tanto a arquitetura do sistema de recomendação original, quanto o estado da arquitetura modificada que seria utilizada para os experimentos, incluindo pseudocódigos que ilustram o funcionamento desta arquitetura e da estratégia original do *baseline*.

Além disso, foram apresentadas as quatro estratégias de prompt propostas pelo trabalho — *Two-step-prompting*, *Single-step-prompting*, *One-shot-prompting* e *Chain-of-thoughts-prompting* — utilizadas em substituição da estratégia *baseline* (*Three-step-prompting*), afim de que se realize a análise comparativa entre todas as prompts avaliadas. Justifica-se ainda, a escolha dos LLMs utilizados nesta análise comparativa, sob a visão de análise da possibilidade de extrair capacidades equiparáveis de recomendação diante do modelo robusto e custoso GPT-3.5-Turbo-Instruct, com modelos abertos e mais acessíveis computacionalmente, sendo em número restrito de parâmetros, com os modelos Llama 3.1 8B Instruct e Gemma-3-4B-IT.

5 Experimentos

Neste capítulo será apresentado o detalhamento dos experimentos realizados, desde as tecnologias utilizadas, códigos fonte, até a análise sobre os resultados obtidos.

5.1 Configuração Experimental

5.1.1 Bases de Dados

Para esta análise comparativa, foram usadas duas bases de dados na execução dos experimentos. A primeira *Amazon Reviews Dataset* (NI; LI; MCAULEY, 2019), consiste num conjunto de dados de análises publicadas por usuários acerca de suas compras de produtos, divididas entre múltiplas categorias que representam subconjuntos diferentes. Foram utilizados aqui os subconjuntos Beauty e Luxury Beauty, com dados dispostos por atributos como id e nome do usuário, id e título do produto analisado e data da *review*.

A segunda base de dados trata-se da *MovieLens 100K* (HARPER; KONSTAN, 2015), um dataset construído e publicado pela organização GroupLens, que representa uma coleção de 100 mil análises de usuários sobre filmes assistidos, dados pelos atributos de *id* do usuário, *id* do item, *rating* (nota de avaliação) e *timestamp*, a varíavel de hora e data da análise. Este é um dataset popular, sendo adotado por diversas pesquisas de sistemas de recomendação (PENG et al., 2025; WU et al., 2024), o que inclui a de Wang e Lim (2023), baseline de comparação da análise feita aqui, além de outros entre os trabalhos relacionados discutidos em tal capítulo.

5.1.2 Ambiente de execução dos experimentos

Um computador pessoal do autor foi usado como ambiente de desenvolvimento e execução dos experimentos realizados. Ele possui como parte da configuração uma GPU Nvidia RTX 4080 com 16GB de memória GDDR5, combinada com a quantidade de 32GB de memória RAM disponíveis para o sistema. Os LLMs abertos foram hosteados usando a capacidade computacional da GPU, com interfaces de requisição das APIs providas pela aplicação Ollama (OLLAMA, 2026), um projeto *open-source* que permite executar instâncias de diversos LLMs abertos, em containers no ambiente local. O modelo da OpenAI utilizado aqui, por sua vez, é provido apenas por APIs públicas de autoria própria, disponíveis sob cobrança por número tokens de entrada

e de saída pela geração. Todo desenvolvimento dos experimentos foi realizado com Python 3.12.

5.1.3 Métricas de Avaliação de Desempenho

Nesta seção serão detalhados as métricas para avaliação de perfomance das recomendações obtidas nos experimentos, que foram utilizadas originalmente pelo baseline, e que vão permitir a comparação entre as estratégias de engenharia de prompt propostas, sendo usadas também na comparação com trabalhos de sistemas de recomendação externos. Portanto, foram utilizadas as métricas *Normalized Distributed Cumulative Gain*@10, *Hit Rate*@10 e *Recall@10*.

5.1.3.1 *Hit Rate*

A *Hit Rate* é uma métrica *Top N* (NAWARA; KASHEF, 2025), referente a fração de acertos que ocorreram nas amostras avaliadas. É usada aqui a métrica *Hit Rate*@K (HR@10), onde $K = 10$, dado que avalia-se o acerto de um item de teste, entre uma seleção de 10 itens da recomendação provida pelo LLM (HE et al., 2017). Havendo 1 item *ground-truth* por amostra, se este item existir na recomendação dada pelo LLM ao usuário-alvo da amostra, configura-se, então um acerto (HE et al., 2015).

O cálculo do Hit Rate@10 pode ser ilustrado pela fórmula de He et al. (2015):

$$HR@K = \frac{\textit{Número de acertos@K}}{|GT|} \tag{5.1}$$

onde $K = 10$ representando o alcance de avaliação entre os 10 itens recomendados, com $|GT|$ se referindo à quantidade de itens *ground-truth*, com $|GT| = 1$ no caso das avaliações feitas aqui.

5.1.3.2 *Normalized Distributed Cumulative Gain* (NDCG)

A métrica, de sigla NDCG, dá a avaliação do rank de relevância atribuído a um item de teste encontrado, por sua vez, num conjunto dado pela recomendação. Avaliando o parâmetro de rank do item *ground-truth*, o cálculo atribui valores entre 0 e 1, onde valores mais próximos de 1 representam maior relevância atribuída ao item avaliado, com 1 sendo o de recomendação perfeita.

- O cálculo segue a fórmula dada também por He et al. (2015):

$$NDCG@K = Z_K \sum_{i=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)}, \tag{5.2}$$

onde Zk representa um normalizador que garante o valor de 1 no ranqueamento perfeito, K representa o total de itens no conjunto da recomendação, i representa o rank do item avaliado e r_i o valor de relevância atribuído ao item candidato, previamente a recomendação.

Nota-se que, para os experimentos feitos aqui, com $K = 10$, existe apenas 1 item *ground-truth* entre os 10 itens produzidos na recomendação e, a relevância atribuída aos itens recomendados é binária, de forma que apenas o item *ground-truth* vai ter a relevância de 1. Desta forma, durante o somatório realizado para cada item da recomendação, todo item que não for o *ground-truth* vai ter valor negativo por ter $r_i = 0$. Como objetivo aqui é de avaliar apenas o rank do item-ground truth quando presente na recomendação, são desconsideradas as avaliações de rank negativas, pois representam que o item-ground truth não estava na recomendação.

Isso implica que, para cada amostra de recomendações avaliada, se o item *ground-truth* estiver no conjunto de itens recomendados, teremos $r_i = 1$ e $2^{r_i} - 1 = 1$, então:

$$NDCG@10 = \frac{1}{\log_2(i+1)}, \tag{5.3}$$

onde i mantem-se referente ao rank posicional do item *ground-truth*.

Portanto, o *NDCG*@10 de uma base de dados avaliada, é dado pela média do NDGC@10 de todas as recomendações avaliadas por amostra do base de dados processada. O snippet A.9 demonstra o código Python usado para medição ambos do valor Hit Rate@10 e do NDCG@10 do conjunto de amostras de uma base de dados.

5.1.3.3 *Recall*

Dado um conjunto de itens de relevância definida, a métrica *Recall*@10 mede a proporção de itens relevantes, do conjunto total, que apareceram numa recomendação (NAWARA; KASHEF, 2025).

A métrica é demonstrada na seguinte fórmula por Nawara e Kashef (2025):

$$Recall@K = \frac{N_{rs@K}}{N_r} \tag{5.4}$$

onde K é o número do alcance de itens a serem avaliados na recomendação, $N_{rs@K}$ é o número de itens relevantes entre os primeiros K itens da lista recomendada e N_r o número total de itens relevantes presentes na recomendação inteira.

Conforme indicado por Kang e McAuley (2018), com $K = 10$ e havendo apenas 1 item de teste, sendo o único item de relevância por recomendação avaliada, temos que a métrica *Hit Rate*@10 é equivalente à métrica *Recall*@10. Isso se dá pois, dado a

fórmula 5.4, com $N_{rs@10} = 1$ item relevante presente, entre o total de 10 itens avaliados e, com $N_r = 1$ do único item relevante definido, temos

$$Recall@10 = \frac{1}{1} \tag{5.5}$$

na presença do item de teste na recomendação, ou, com $N_{rs@10} = 0$,

$$Recall@10 = \frac{0}{1} \tag{5.6}$$

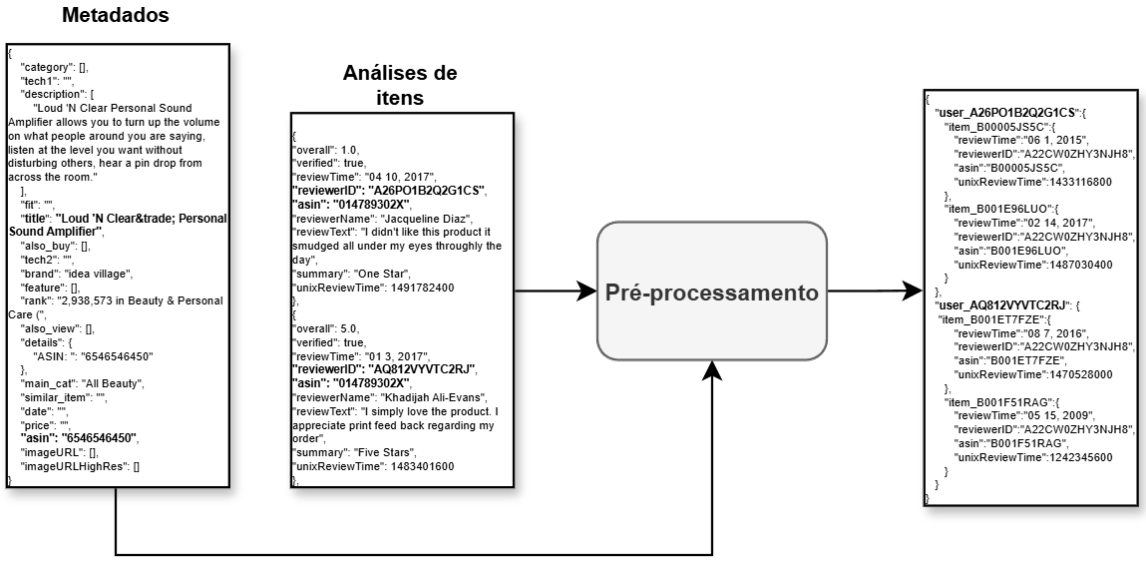
quando o item de teste não está presente na recomendação, de modo que o *Hit Rate*@10 avaliado aqui se dá também mediante presença do item *ground-truth*, sendo possível esses mesmos valores binários.

Dessa forma, tanto o *Recall*@10 e o *Hit Rate*@10 de uma base de dados avaliada aqui, é dado pela média de todas as medições, entre todas as amostras avaliadas do conjunto de amostras do *dataset*. Este valor de *Recall*@10 é usado como parte dos resultados obtidos nos experimentos com o *dataset* MovieLens 100K, onde se compara os valores de *Recall@10* e *NDCG@10* com os encontrados em trabalhos externos de sistemas de recomendação. O snippet A.9 no apêndice mostra a codificação para cálculo dessas métricas.

5.2 Criação das amostras

5.2.1 Pré-processamento de Dados

Figura 15 – Esquema do pré-processamento



Fonte: o Autor

Para criação das amostras, foi necessário, primeiramente, realizar o pré-processamento dos dados brutos nas duas instâncias dos experimentos realizados. Nessa

etapa, de esquema geral ilustrado na figura 15, o objetivo é de organizar os dados de itens de usuários, numa coleção estruturada onde usuários tem os itens de suas análises relacionados a si, de forma que seja possível consultar quais são os itens por usuário, e quais são as informações de cada item relacionado. São informações de relacionamento que serão utilizadas em etapas seguintes da criação de amostras.

Os conjuntos Beauty e Luxury, *subsets* da base de dados de análises Amazon, foram escolhidos dado algumas particularidades. Primeiro, a escolha foi motivada a partir do tamanho do subset, de modo que fosse viável, diante das restrições computacionais, a leitura e filtragem dos registros do conjunto. Por último, a motivação era de que houvesse um espaço de conectividade modestamente denso na relação entre usuários e itens. Isso foi decidido para permitir variedade de itens, que fossem avaliados por quantidades maiores de usuários, o que torna relações de coocorrência de itens interagidos entre usuários diferentes mais ampla, algo desejável quando se quer estipular relações de popularidade de interação dos itens. Em dados não-processados, então, no conjunto Beauty tratam-se de 371.345 análises sobre 32.992 produtos, enquanto no conjunto Luxury Beauty, tratam-se de 574.628 análises sobre 12.308 produtos. O snippet A.8 mostra a implementação do pré-processamento no apêndice A.

Um outro aspecto da utilização desse dataset, envolveu uma “pré-filtragem” de usuários por quantidade de itens interagidos, implementada pelo código no snippet 5.1. Isso foi necessário pois, para a estruturação das prompts de recomendação, foi decidido que os usuários deveriam ter pelo menos o mesmo número de itens itens interagidos que o número de itens requisitados na recomendação, sendo aqui 10 itens. A escolha deu-se para que a análise pudesse ser em cima de amostras onde conjuntos de histórico de interações dos usuários alvo não fosse tão pequena, dado a constatação de muitos registros que tinham pouquíssimas interações. Dessa forma, as amostras se dividem para 22 usuários-alvo do conjunto Beauty e 830 usuários-alvo do conjunto Luxury Beauty.

Snippet 5.1 – Snipet da filtragem por número de interações do usuário

```
1  def get_k_user_reviews(k, user_reviews_dict):
2      """ Retorna um dicionário user-reviews, com número mínimo k de reviews por
       usuário, entre todas as reviews"""
3
4      k_reviews_user = {}
5      product_subset = {}
6      product_count = 0
7      for user, reviews_dict in user_reviews_dict.items():    # Para cada usuário
8          if len(reviews_dict) >= k:  # Contabiliza seu número de análises
9              k_reviews_user[user] = reviews_dict
10
11             for asin, review in reviews_dict.items():   # Contabiliza número de
       produtos de todos usuários que tem pelo menos k análises
12                 if asin not in product_subset:
13                     product_subset[asin] = 0
```

```
14                product_subset[asin] += 1
15
16    return k_reviews_user, product_subset   # Retorna os usuários que tem k ou mais
      reviews, e um conjunto de seus produtos
```

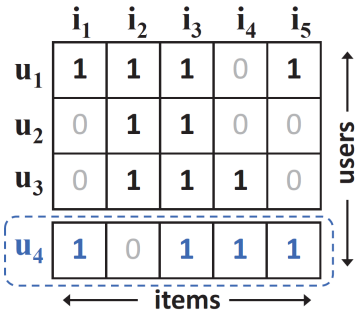
Os dados brutos do *dataset* MovieLens 100K (GROUPLENS, 2026), passam pelo mesmo pré-processamento que envolve carregar os dados em uma coleção e torná-la passível de consulta, para que sejam selecionados os itens de interação dos usuários. Não foi necessário, no caso desta base de dados, uma "pré-filtragem" de usuário por quantidade de itens interagidos, pois aqui cada usuário tem, pelo menos, 20 análises, que contam como interações com os itens. Os 100.000 registros de análises de filme são, portanto, de autoria divida entre 943 usuários sobre 1682 itens.

5.2.2 Filtragem por usuários

A criação das amostras se dá pela filtragem por usuários, num processo que busca decidir quais itens são os mais semelhantes às preferências do usuário alvo. Isso é feito mediante comparação dos itens interagidos do usuário alvo com os itens interagidos de todos os outros usuários da base de dados. A seguir detalha-se o algoritmo fornecido pelo baseline, dado por Wang e Lim (2023):

- cada vetor binário de interações dos usuários, cujas posições representam um item do histórico, vão compor primeiramente uma matriz que chamamos de *item-hot*. Chama-se aqui *item-hot* ou *user-hot* os vetores binários que indicam a presença de itens ou usuários, pois consistem em representações *one-hot*, onde *features*, referentes aqui a itens ou usuários, são dispostas posicionalmente e têm a sua presença marcada com valor positivo na devida coluna (HE et al., 2017), conforme figura 16;

Figura 16 – Exemplo de matriz usuario-item



(a) user–item matrix

Fonte: (HE et al., 2017)

- a seguir, é obtido o resultado do produto entre a matriz *item-hot* e ela mesma transposta. Quando transposta, a matriz *item-hot* do usuários se torna *user-hot*,

significando que cada linha representa um vetor binário de usuários que tiveram interação com o item. O produto das matrizes *item-hot* e *user-hot* vai, portanto, resultar numa matriz *user x user* de coocorrências de mesmos itens consumidos entre usuários, onde em cada linha encontra-se um vetor com somatórios das coocorrências de interações que um usuário tem com todos os outros. A imagem 19 no apêndice A mostra a matriz *user x user* obtida para o subconjunto Amazon Beauty.

Snippet 5.2 – Função de cálculo somatório das coocorrências de interações em itens que os usuário tem entre si

```
def user_similarity(user_reviews_dict, product_subset_len, subset_products_list):
'''
Esta função para filtragem por usuários segue algoritmo implementado originalmente por Wang e Lim (2023), e retorna matriz de co-ocorrências de produtos interagidos entre todos usuários, sendo uma matriz usuário-usuário
'''

    user_items_hot_matrix = []  # matriz de vetores item-hot dos usuários

    for user_id_key, dict_value in tqdm(user_reviews_dict.items(), total=len(user_reviews_dict.values()), desc="Computing user-similarity"):

        user_items_hot_vect = np.zeros([product_subset_len], dtype=np.float64)

        for product_id_key, review_value in dict_value.items(): #   Cria vetor one-hot usuario-produto
            product_vector_position = get_product_pos(review_value['asin'], subset_products_list)

            user_items_hot_vect[product_vector_position] = 1

        user_items_hot_matrix.append(user_items_hot_vect)   # Cria matriz de vetores one-hot usuario-produto

    user_item_hot_matrix = np.array(
        user_items_hot_matrix)  # Matrix user-item, com Vetor binário de itens consumidos, um item por coluna, um vetor referente a um usuário por linha
    user_item_hot_matrix_transposed = user_hot_matrix.transpose()  # Matrix inversa user-item, temos item por usuários, vetor binário de usuários consumidores, um vetor referente a um item por linha
    user_user_ocurr_mat = np.dot(user_hot_matrix,
                                 user_hot_matrix_transposed)  # Matrix user-user, por linha um vetor com o somatório de ocorrências iguais de itens consumidos, por usuário com outros usuários

    return user_item_ocurr_mat
```

- a partir da matriz de coocorrências de itens interagidos entre usuários, será feito, para cada usuário alvo a ser definido na amostra, a seleção dos itens candidatos. Dado um vetor *user-user* de um usuário alvo, são escolhidos os usuários candida-

tos os quais tiveram maiores números de coocorrências de interações. Para cada usuário-candidato respectivo, calcula-se um valor de peso por similaridade, dado pelo produto do total de coocorrências entre usuário-alvo e o usuário-candidato com o valor de 1 divido pelo somatório de todas coocorrências do usuário-alvo, seguindo a fórmula:

$$Peso = Ou_i \times \frac{1}{\sum_{j=1}^{n} Ou_j} \tag{5.7}$$

, onde Ou_i é total coocorrências entre usuário-alvo e usuário-candidato, e o somatório $\sum_{j=1}^{n} Ou_j$ sendo de todas coocorrências entre usuário-alvo e os usuário-candidatos.

São selecionados os itens dos usuários-candidatos que não estão no conjunto de interações do usuário-alvo, atribuindo para cada item selecionado um valor que se incrementa pelo peso sempre que indicado repetidamente pelos usuários-candidatos. Essa seleção representa, então, os itens mais populares entre o usuário-alvo e os usuários-candidatos. Finalmente, realiza-se uma última seleção dos itens de maior valor de popularidade, dado um número de itens candidatos definido, que encerra a filtragem por usuários.

O snippet 5.3 contém a implementação do algoritmo que calcula os valores pesos-similaridade entre os itens do usuário-alvo e usuário-candidatos, e a tabela 8 no apêndice A mostra o exemplo de uma relação dos maiores valores peso-similaridade calculas para itens entre um usuário-alvo e os usuários-candidatos. A partir desta relação é selecionado o conjunto de itens candidatos.

Snippet 5.3 – Função de cálculo das similaridades entre os usuários sobre seus itens

```
1  def get_user_filtering_candidates(user_reviews_dict, target_user_items,
       user_ocurr_vec, num_user_candits, num_item_candits):
2  '''
3  Função de filtragem por usuários, implementação feita para o contexto de variáveis
       deste projeto, mas segue implementação originalmente foi feita para o baseline
       por Wang e Lim (2023).
4  A função filtra os usuários que tiveram maior número de coocorrências com o usuário
       alvo, calcula o valor de peso-similaridade dos itens dos usuarios-candidatos, e
       escolhe os itens de maior valor de peso para retorno do conjunto de itens
       candidatos.
5  '''
6      sorted_user_ocurr_mat = sorted(list(enumerate(user_ocurr_vec)), key=lambda x: x
       [-1], reverse=True)[
7                              :num_user_candits]
8      user_ocurr_sum = sum([e[-1] for e in sorted_user_ocurr_mat])
9
10     candidate_items_dict = {}
11     for user_i, user_ocurr_val in sorted_user_ocurr_mat:    # Iteração no Vetor
       ordenado de Similaridade que um usuário tem com outros usuários, vindo do
       produto da matrix user-item (item-hot) com ela mesma transposta
12         weighted_user_siml = user_ocurr_val * 1.0 / user_ocurr_sum
```

```
        user_items = list(user_reviews_dict.values())[user_i]

        for item_id in user_items:  # Construção do conjunto de itens candidatos
            if item_id not in target_user_items:    # Se o item do usuário candidato não estiver no conjunto de itens do usuário-alvo
                if item_id not in candidate_items_dict:
                    candidate_items_dict[item_id] = 0.
                candidate_items_dict[item_id] += weighted_user_siml # para cada item do conjunto de itens do usuário por iteração, incrementado o valor de similaridade/"proximidade" nessa inicial lista de itens candidatos

    '''
    Ao fim do laço, terá sido construído um conjunto de itens candidatos, respectivamente com seus valores de similaridade somados, de todos usuários candidatos.
    Então o dict montado representa, para cada item, seu valor de proximidade entre os itens de interesse entre usuário alvo e a média os seus usuários candidatos mais similares.
    A ser visto como conjunto de itens mais "populares" para recomendação ao usuário alvo.
    '''
    candidate_pairs = list(sorted(candidate_items_dict.items(), key=lambda x: x[-1], reverse=True))
    candidate_items = [cleanse_item_title(metadata_dict[e[0]]['title']) for e in candidate_pairs][:num_item_candits]
    return candidate_items
```

5.2.3 Definição de item *Ground-Truth*

Para escolha do item de teste usado para avaliar a performance das recomendações, segue-se a estratégia *leave-one-out*, também utilizada por Wang e Lim (2023) e implementada pelo Autor com o código do snippet 5.4. Nela, o item de interação mais recente é o escolhido como *ground-truth*, que seria o item de interação seguinte do usuário alvo dado seu histórico de interação (XIE et al., 2021). Este item de teste é retirado do conjunto de interações do usuário-alvo, para ser incluído entre seus itens candidatos. É o desejo, portanto, que esse item esteja entre os itens recomendados pelo LLM, de forma que será avaliado tanto sobre sua presença, quanto sobre em que *rank* o item ground-truth foi colocado.

Snippet 5.4 – Função da escolha do item ground-truth

```
def ground_truth_item(user_reviews_dict):

'''
Esta função seleciona 1 item para se definir como ground-truth de teste na amostra do usuário-alvo, escolhendo o item de interação mais recente.
Retorna o item ground-truth e o conjunto de itens interagidos do usuário sem o ground-truth
'''
    date_format = '%m %d, %Y'   # Formato de data encontrado no registro

    bigger_date = None
```

```
    ground_truth_item = None
    train_reviews_dict = user_reviews_dict.copy()

    for asin_key, review_value in user_reviews_dict.items():    # Laço que define o item mais recente do usuário

        review_date = datetime.datetime.strptime(review_value['reviewTime'], date_format)

        if bigger_date is None:
            bigger_date = review_date

        if review_date >= bigger_date:
            bigger_date = review_date
            ground_truth_item = review_value

    train_reviews_dict.pop(ground_truth_item['asin'])   # Retira o item ground_truth do conjunto de itens interagidos do usuário, para ser colocado entre os itens candidatos
    return ground_truth_item, train_reviews_dict
```

5.2.4 Formatação dos templates das estratégias

Cada estratégia foi definida numa *string* passível de formatação, coletadas num arquivo de extensão json. Esse arquivo é carregado no componente de montagem e estratégias de prompts, e a formatação de cada estratégia ocorre imediatamente antes de serem enviadas na requisição ao LLM. Foram criadas funções que concatenam as repostas do LLM adequadamente nos casos das estratégias com múltiplas promts, que são as estratégias *three-step-prompting* dada pelo *baseline* e *two-step-prompting* proposta pelo Autor.

As outras três estratégias, *single-step-promptin*, *one-shot-prompting* e a *chain-of-thoughts-prompting*, consistem em apenas uma prompt cada e, todas as prompts, de todas as estratégias, vão ser formatadas com a inclusão de cada definido conjunto de itens candidados e itens de interação do usuário-alvo, por amostra.

No caso da estratégia one-shot, era necessário incluir um exemplo de recomendação, estruturados com os itens de interação e itens candidatos, como contexto da tarefa de recomendação solicitada ao LLM. Foi planejado, então, inserir exemplos de recomendações dadas a amostras que já tinham sido utilizadas em outras execuções, de outras estratégias. Mais especificamente, foi decidido utilizar exemplos que vinheram da execução com a estratégia *three-step-prompting* dada pelo baseline, a fim de investigar o impacto de desempenho ao usar recomendações obtidas por essa estratégia, de prompt múltiplas. Para os experimentos na base de dados Amazon, foram escolhidos dois exemplos onde houve acerto do item teste na recomendação, de forma que os exemplos não se duplicassem nas prompts formatadas de mesmas amostras. Para a estratégia one-shot na base de dados MovieLens 100K, também foi escolhido um

exemplo onde houve acerto do item de teste, na execução *three-step-prompting*, para uma amostra que não se repete entre as separadas para a execução na nova estratégia. As prompts one-shot eram formatadas, portanto, de duas maneiras a seguir, onde duas amostras iguais não se repetem em exemplo e itens para inferência. Exemplos das prompts de todas estratégias, formatadas, são encontradas no Apêndice A.

O snippet 5.5 mostra a estrutura da coleção de amostras para um *dataset* processado, a ser carregado no módulo de montagem de prompts, que iterativamente, por usuário-alvo, formata as prompts com os itens e os envia para o LLM. Foram definidos os atributos *revd*, de itens de interação do usuário-alvo, *canddts* dos itens candidatos para a recomendação e *tst_canddt* para o item *ground-truth*.

Snippet 5.5 – Exemplo de amostra de um dataset processado

```
{
    "A1YY53NQXFKMRN": {
        "revd": [
            "Crabtree & Evelyn Gardener's Ultra-Moisturising Hand Therapy Pump 250g/8.8 OZ",
            "Revision Skincare Nectifirm 1.7 oz",
            "NEOCUTIS Lumière Bio-restorative Eye Cream 0.5 Fl Oz",
            "AHAVA Dead Sea Dermud Intensive Moisturizers",
            "AHAVA Dead Sea Mineral Body Lotions",
            "AHAVA Face & Body Essentials Starter Kit Velvet Cream Wash Mineral Hand Cream
                Purifying Mud Mask",
            "Crabtree & Evelyn Gardeners Ultra-Moisturising Hand Cream Therapy 3.5 oz",
            "EltaMD UV Daily Tinted Facial Sunscreen Broad-Spectrum SPF 40 1. 7 oz",
            "Colorescience Tint du Soleil SPF 30 UV Protective Foundation",
            "SkinMedica HA5 Rejuvenating Hydrator",
            "NEOCUTIS Bio Serum"
        ],
        "canddts": [
            "SkinMedica AHA/BHA Exfoliating Cleanser 6 oz.",
            "Dermablend Quick-Fix Full Coverage Concealer 0.16 Oz.",
            "Dermablend Cover Creme Full Coverage Foundation with SPF 30 1 Oz",
            "Vichy Puret Thermale One Step Cleanser for Sensitive Skin 3.3 Fl. Oz.",
            "Dermablend Intense Powder High Coverage Foundation",
            "Dermablend Smooth Liquid Foundation with SPF 25 1 Fl. Oz.",
            "Dermablend Quick-Fix Body Makeup Full Coverage Foundation Stick 0.42 Oz.",
            "AHAVA Dead Sea Mineral Hand Creams",
            "Vichy Pureté Thermale One Step Micellar Cleansing Water",
            "SkinMedica TNS Essential Serum 1 oz.",
            "Vichy Aqualia Thermal Rich Cream Moisturizer",
            "SKIN&CO Roma Truffle Therapy Eye Concentrate 0.5 fl. oz.",
            "Obagi Professional-C Serum 1 fl. oz.",
            "iS CLINICAL  Youth Intensive Crème 1.7  Oz",
            "NuFACE Anti-Aging Infusion Serum | Use with NuFACE Device",
            "La Roche-Posay Toleriane Teint Mattifying Mousse Foundation 1 Fl. Oz.",
            "Erno Laszlo Hydra-Therapy Memory Sleep Mask 1.35 fl. oz.",
            "Obagi Nu-Derm Clear Fx 2 oz.",
            "TIZO 2 Non-Tinted Facial Mineral Sunscreen SPF 40  1.75 oz",
            "La Roche-Posay Anthelios Tinted Mineral Primer with SPF 50 1.35 Fl. Oz."
        ],
        "tst_canddt": "SkinMedica AHA/BHA Exfoliating Cleanser 6 oz."
    }
}
```

5.3 Resultados e Análise

São apresentados, nesta seção, os resultados dos experimentos de recomendação com LLMs. Também são apresentadas análises, que buscam discutir os aspectos das estruturas de engenharia de prompt propostas, sob ótica das capacidades providas pelos LLMs GPT-3.5-Turbo-Instruct, Llama-3.1 8B Instruct e Gemma-3-4B-IT, a partir dos valores de desempenho medidos pelas métricas *Hit Rate@10*, *Recall@10* e *NDCG@10*.

Os experimentos foram realizados em duas partes. Na primeira parte, testou-se o desempenho de recomendação com as amostras criadas a partir da base de dados Amazon Reviews Dataset detalhada em seções anteriores. As 4 estratégias de engenharia de prompt propostas, nomeadas *Two-step-prompting*, *Single-step-prompting*, *one-shot-prompting* e *Chain-of-thoughts-prompting*, foram aplicadas em substituição à estratégia originalmente proposta por (WANG; LIM, 2023), nomeada *Three-step-prompting*. Sequencialmente, cada estratégia foi usada para execução dos experimentos de recomendação, onde todas solicitam ao LLM a recomendação de 10 itens, dado cada conjunto de itens candidatos em cada amostra separada por usuário-alvo de recomendação. Coletados as recomendações de todas as amostras, para cada uma das estratégias executadas em combinação com cada LLM, foi realizado a contabilização das métricas.

Na segunda parte, foi escolhida apenas uma das 4 estratégias de prompt propostas, mediante comparação de seu desempenho obtido na parte anterior, para que sejam executados novos experimentos, na base de dados MovieLens 100K. São nestes onde se compara não somente o desempenho obtido entres as 3 LLMs usadas, mas também entre trabalhos externos que apresentam sistemas de recomendação de naturezas diferentes. Tais trabalhos foram escolhidos pela motivação de que também utilizavam o mesmo *dataset* MovieLens 100K, então a comparação feita nessa parte vai englobar os valores de *performance* obtidos com a estratégia do baseline, com a estratégia proposta escolhida e com valores de outros dois sistemas de recomendação, encontrados em publicações externas.

5.3.1 Resultados dos experimentos no Amazon Reviews Dataset

Foram aferidas as métricas de desempenho *HitRate@10* e *NDCG@10* das 5 estratégias aplicadas, entre os 3 LLMs usados, representando as médias de cada métrica entre 22 amostras no dataset Amazon Beauty, e 830 amostras no dataset Amazon Luxury Beauty. As estratégias “vitoriosas” foram variadas entre os LLMs que proveram as recomendações.

Nos experimentos de valores dispostos na tabela 2, com o modelo GPT-3.5-

Turbo-Instruct, o mais poderoso entre os modelos avaliados, notou-se que a estratégia vitoriosa entre os subsets foi consistente em conseguir tanto maior número de acertos quanto maior valor de relevância atribuído ao item de teste. Constata-se também que, entre os dois subsets, as estratégias *Two-step-prompting* e *one-shot-prompting*, propostas neste trabalho, conseguiram desempenho maior que a estratégia do baseline *Three-step-prompting*.

Tabela 2 – Resultados dos experimentos na base de dados Amazon Reviews Dataset com o modelo GPT-3.5-Turbo-Instruct

Dataset	**Estratégia**	HitRate@10	NDCG@10
Amazon Beauty	Three-step-prompting	0.3636	0.2727
	Two-step-prompting	**0.6363**	**0.3755**
	Single-step-prompting	0.4090	0.2237
	one-shot-prompting	0.5000	0.2422
	Chain-of-thoughs prompting	0.5454	0.2441
Amazon Luxury Beauty	Three-step-prompting	0.4337	0.1948
	Two-step-prompting	0.3783	0.1718
	Single-step-prompting	0.3855	0.1718
	one-shot-prompting	**0.4397**	**0.2294**
	Chain-of-thoughts-prompting	0.3939	0.1944

Fonte: o Autor

Os resultados obtidos pelas execuções com o modelo Llama-3.1 8B Instruct, encontrados na tabela 3, mostraram desempenho próximo ao obtido com o modelo da OpenAI, entretanto inconsistentes nas estratégias de prompt vitoriosas. Destaca-se a vitória da estratégia do baseline no *subset* Amazon Beauty, bem como o melhor desempenho de *HitRate@10* em empate com a estratégia *Chain-of-thoughts* proposta e, ambos valores foram superiores aos obtidos pelo GPT-3.5-Turbo-Instruct. Já para o *subset* Amazon Luxury Beauty, é visto novamente as duas estratégias se saindo melhor mas, em métricas diferentes, com a *Three-step-prompting* sendo superior na taxa de acertos nessa amostragem maior de recomendações, enquanto a estratégia de técnica *chain-of-thoughts* atribuiu relevância maior ao item *ground-truth* por mais vezes, entretanto, nenhum dos melhores valores obtidos com o modelo da Meta superaram os do modelo anterior.

Com o uso do modelo Gemma-3-4B-IT, o modelo de menor número de parâmetros entre os avaliados, para a recomendação de itens do *subset* Amazon Beauty a estratégia *Single-step-prompting* conseguiu ambos melhores valores de desempenho, havendo um empate do valor *HitRate@10* de 0.4090 entre todas as estratégias, exceto a *Two-Step-prompting*. O valor de empate foi próximo aos valores medidos dos outros dois modelos, já o melhor valor de atribuição de relevância, dado pelo *NDCG@10*

de 0.2472, foi o pior entre os 3 modelos avaliados. Para o *subset* Amazon Luxury Beauty, os valores mais altos em *HitRate@10* e *NDCG@10*, obtidos pela estratégia *chain-of-thoughts*, foram os menores entre todos os valores *vencedores* nas avaliações. Destaca-se, por fim, os piores desempenhos obtidos em ambas métricas, entre todos os modelos avaliados, com *HitRate@10* de 0.1614 e *NDCG@10* de 0.0935 pela estratégia *Two-step-prompting*. Seus resultados estão listados na tabela 4 e a figura 17 apresenta um gráfico da comparação entre todas as combinações modelo-estratégia avaliadas.

Tabela 3 – Resultados dos experimentos na base de dados Amazon Reviews Dataset com o modelo Llama-3.1 8B Instruct

Dataset	**Estratégia**	HitRate@10	NDCG@10
Amazon Beauty	**Three-step-prompting**	**0.6818**	**0.4947**
	Two-step-prompting	0.5000	0.3017
	Single-step-prompting	0.5000	0.3036
	one-shot-prompting	0.3636	0.2783
	Chain-of-thoughs prompting	**0.6818**	0.4310
Amazon Luxury Beauty	Three-step-prompting	**0.3891**	0.1757
	Two-step-prompting	0.3855	0.1442
	Single-step-prompting	0.3530	0.1741
	one-shot-prompting	0.2915	0.1738
	Chain-of-thoughts-prompting	0.3626	**0.1880**

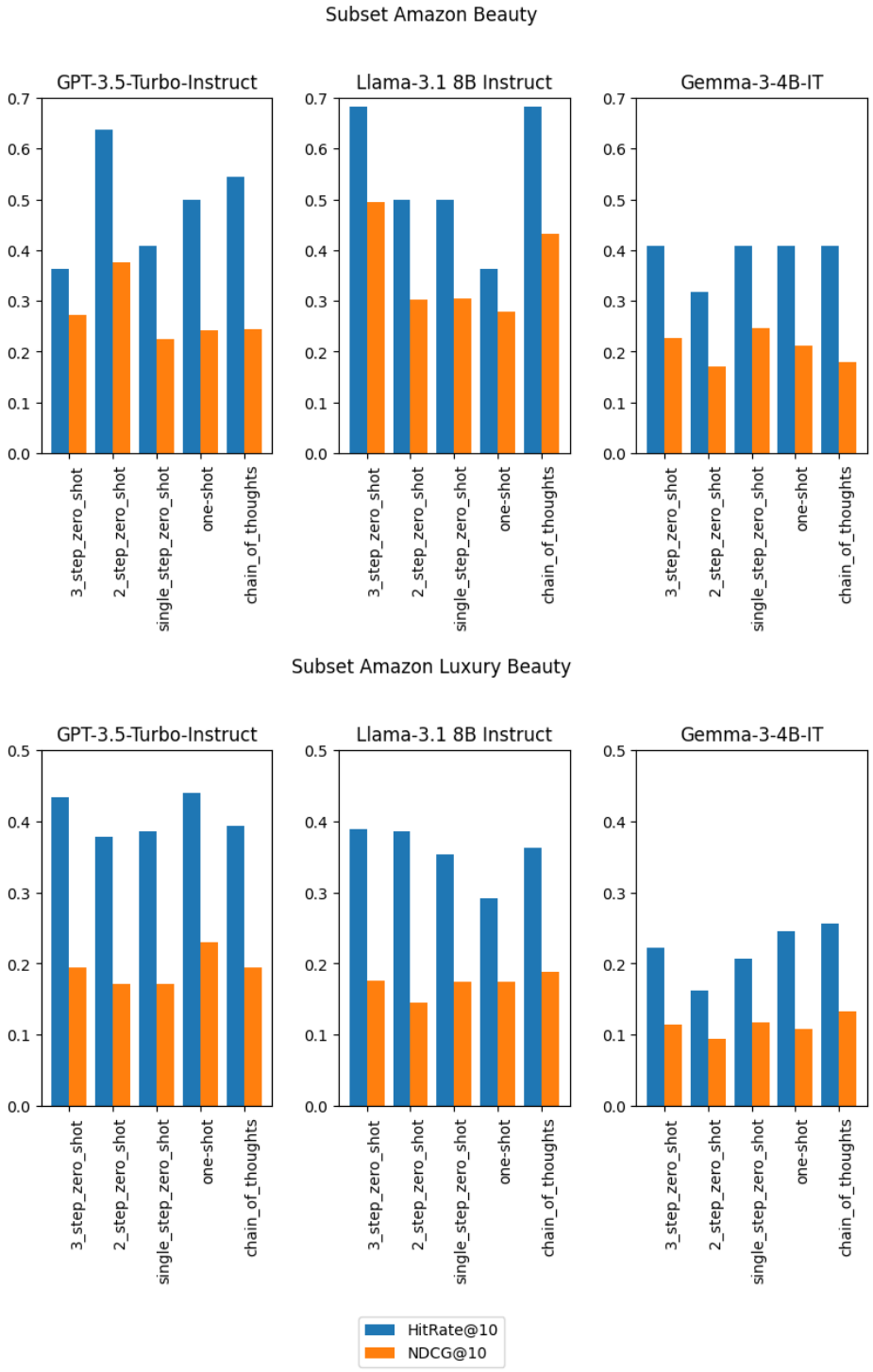
Fonte: o Autor

Tabela 4 – Resultados dos experimentos na base de dados Amazon Reviews Dataset com o modelo Gemma-3-4B-IT

Dataset	**Estratégia**	HitRate@10	NDCG@10
Amazon Beauty	Three-step-prompting	**0.4090**	0.2258
	Two-step-prompting	0.3181	0.1695
	Single-step-prompting	**0.4090**	**0.2472**
	One-shot-prompting	**0.4090**	0.2110
	Chain-of-thoughs prompting	**0.4090**	0.1789
Amazon Luxury Beauty	Three-step-prompting	0.2216	0.1141
	Two-step-prompting	0.1614	0.0935
	Single-step-prompting	0.2060	0.1169
	One-shot-prompting	0.2457	0.1077
	Chain-of-thoughts-prompting	**0.2554**	**0.1318**

Fonte: o Autor

Figura 17 – Comparação das estratégias nos subconjuntos do dataset de análises Amazon



Fonte: o Autor

5.3.2 *Ablation Study*: Impacto dos aspectos das prompts

Propõe-se um estudo de ablação para análise dos aspectos das prompts que podem ter causalidade de impacto no desempenho da tarefa de recomendação com LLMs. Foram definidos em componentes, que explicam a presença de aspectos espe-

cíficos encontrados entre as estratégias de engenharia de prompt avaliadas:

- **Feedback Implícito do Usuário:** sobre a presença, na prompt, do contexto de itens que o usuário interagiu e que servem como "entendimento" sobre quais tipos, variedades, gêneros e/ou classificação de itens o usuário tem interesse — em essência, trata-se do histórico de itens interagidos pelo usuário. Esse componente é presente em todas estratégias de prompt.

- **Fornecimento de Exemplo:** sobre a inclusão, na prompt, de um exemplo de como deve ser gerada a recomendação pela LLM. É parte da estratégia de conclusão pelo LLM, elicitada através da prompt, nesse caso na estratégia one-shot. É incluída na prompt o conjunto de itens candidatos, itens interagidos, e a lista de recomendação que inclui o item *ground-truth*, no formato esperado e descrito na prompt.

- **Preferencias do Usuário:** sobre incluir na prompt, uma descrição das preferências do usuário acerca dos itens que foram interagidos, de forma a representar seus "gostos", de acordo com o histórico de itens interagidos. É o aspecto encontrado nas estratégias de prompts múltiplas, que inclui a estratégia dada pelo baseline (WANG; LIM, 2023), de 3 etapas, e a estratégia proposta pelo Autor, com menos *tokens* e com 2 etapas. Nestas estratégias, um dos passos de prompt solicita ao LLM a inferência sobre as preferências do usuário a partir de seus itens interagidos, para fornecer tal descrição na prompt que solicita a inferência da recomendação de itens, informando contexto sobre os tipos de itens que mais se alinham com as preferências.

- **Elicitação de Preferências:** sobre pedir ao LLM que faça a recomendação de acordo com um entendimento de preferências do usuário, de acordo com os itens interagidos, de forma inerente aos parâmetros que o modelo possui e sem a necessidade de prompts múltiplas. Desta forma, a prompt que solicita a inferência da recomendação faz isso pedindo, explicitamente na prompt, que a recomendação seja feita de acordo com as preferências do usuário. Dessa maneira, buscase extrair do LLM o potêncial de inferir as preferências do usuário, ao mesmo que tempo que estas sejam utilizadas de forma inerente às *features* do LLM, para realizar a inferência das recomendações.

A tabela 5 dispõe a relação dos aspectos que se constituem as estratégias de engenharia de prompt avaliadas, listando para cada estratégia, respectivamente, os melhores valores de desempenho de recomendação obtidos para os experimentos no

Tabela 5 – *Ablation study* dos aspectos das prompts com melhores resultados das execuções para o *subset* Amazon Luxury Beauty

Estratégia	Feedback Implícito do Usuário	Fornecimento de Exemplo	Preferencias do Usuário	Elicitação de Preferências	HR@10	NDCG@10
three-step prompting	✓	–	✓	–	0.4337	0.1948
two-step prompting	✓	–	✓	–	0.3855	0.1718
single-step prompting	✓	–	–	✓	0.3855	0.1718
one-shot prompting	✓	✓	–	✓	**0.4397**	**0.2294**
chain-of-thoughts prompting	✓	–	–	✓	0.3939	0.1944

Fonte: o Autor

subset Amazon Luxury Beauty da base de dados Amazon Reviews Dataset. Em primeira instância, nota-se a presença do aspecto de “Feedback Implícito do Usuário” em todas as estratégias de prompt avaliadas, e esta é considerada aqui como o contexto mínimo a ser fornecido ao LLM, para que ele, por sua vez, realize a tarefa de recomendação. Analisando a presença dos aspectos ao redor das estratégias, é possível constatar a causalidade que eles tiveram no desempenho obtido, a depender da técnica de engenharia de prompt que a estratégia aplicou.

Comparando as duas estratégias que aplicam a técnica *zero-shot* e que possuem o mesmo aspecto de “Preferências do Usuário”, *three-step-prompting* e *two-step-promting*, nota-se que a de 3 passos conseguiu resultado superior. Apesar de ambas incluírem as preferências do usuário-alvo, isso foi representado de maneiras diferentes, com a prompt *two-step* de recomendação passando uma lista de itens da preferência do usuário-alvo, inferida no primeiro passo pelo LLM. Já na prompt *three-step*, se infere uma descrição das *features* que representam as preferências do usuário, que é fornecida na prompt de recomendação. Na outra estratégia feita sob a técnica *zero-shot*, não se faz presente aspecto de fornecimento das preferências do usuário, possuindo, em seu lugar, o aspecto de “Elicitação de Preferências”. Com ela, obteve-se um resultado idêntico ao da estratégia de dois passos, mas inferior ao de três passos e, a partir disso, pode-se destacar a importância do fornecimento do contexto de preferências do usuário-alvo quando se aplica a técnica *zero-shot* de engenharia de prompt.

As outras duas estratégias representam técnicas diferentes de engenharia de prompt, com seus aspectos sendo não encontrados na maioria das outras estratégias *zero-shot*. Comparando a técnica *chain-of-thoughts* com a única das estratégias *zero-shot* onde também se caracteriza a “Elicitação de Preferências”, avalia-se que a pri-

meira técnica foi superior nesse aspecto, sob um ganho de 2.17% no HitRate@10 e ganho de 13.15% no NDCG@10. A estratégia *one-shot* é a única caracterizada pela inclusão de exemplo de recomendação prévia e, foi a estratégia que obteve desempenho superior a todas outras estratégias. Em combinação com o aspecto de elicitar preferências do usuário-alvo, a inclusão de exemplo, portanto, foi o aspecto de causalidade dos melhores resultados obtidos. Salienta-se que, outras estratégias que também objetivaram elicitar do LLM as preferências do usuário-alvo, ao mesmo tempo que solicitavam recomendação, tiveram resultados inferiores.

Por fim, se reconhece que a estratégia dada pelo *baseline*(WANG; LIM, 2023) foi superada apenas pela estratégia *one-shot* e é evidenciada a eficiência que a técnica *zero-shot* pode ter na tarefa de recomendação. Sua execução é dada em múltiplas etapas, que são realizadas com o objetivo de extrair o contexto de predileção que um usuário-alvo tem acerca de itens (neste caso, produtos consumíveis). Esse contexto quando informado é, portanto, de alto valor para que a técnica *zero-shot* tenha bom desempenho na tarefa de recomendação com LLMs. Ainda assim, aponta-se os ganhos de desempenho da técnica *one-shot* sobre a *zero-shot*, sendo de 1.38% superior no HitRate@10 e de 17.76% a mais no NDCG@10.

5.3.3 Comparação com trabalhos externos

Como segunda parte da avaliação da tarefa de recomendação com LLMs, foram executados os experimentos de recomendação em cima da base de dados MovieLens 100K. Nesta rodada, foram usadas apenas as estratégias *one-shot-prompting* e *three-step-prompting*. A estratégia *one-shot* foi a escolhida devido aos resultados de *performance* obtidos nos experimentos com a base de dados de análises de produtos Amazon, sendo esta a estratégia de resultados superiores entre todas avaliadas.

Seguindo a pretensão de comparação desta análise, foi executado também a estratégia *three-shots-prompting*, que provêm do trabalho de Wang e Lim (2023). Reiterando a citação encontrada múltiplas vezes entre os capítulos deste trabalho, tal estratégia é o *baseline*, ou parâmetro de comparação aqui e, desta forma, analisa-se como a estratégia *one-shot* proposta desempenha diante da estratégia base, usando o *dataset* de avaliação publicado originalmente. Ainda além, o propósito é de analisar como ambas estratégias, *baseline* e *one-shot* proposta, desempenham como sistemas de recomendação, em comparação a sistemas apresentados em trabalhos externos.

Os sistemas de recomendação externos, escolhidos para esta comparação, foram originalmente avaliados em cima da mesma base de dados MovieLens 100K, sob métricas também encontradas nas avaliações feitas neste trabalho. Tratam-se de sistemas que usam as capacidades dos LLMs como recomendadores. A tabela 6 elenca os resultados de desempenho *Recall@10* e *NDCG@10* obtidos entre os sistemas en-

volvidos nesta comparação. Os resultados dos trabalhos de Ebrat et al. (2025) e Liang et al. (2025) foram extraídos diretamente de suas respectivas publicações.

Tabela 6 – Resultados dos experimentos na base de dados MovieLens 100K em comparação com sistemas externos

Sistema de recomendação	**Recall@10**	**NDCG@10**
Three-step-zero-shot-NIR (WANG; LIM, 2023)	0.4644	0.2085
One-shot GPT	0.4306	0.1802
One-shot Llama	0.2841	0.1085
One-shot Gemma	**0.4803**	0.1581
(EBRAT et al., 2025)	0.2173	**0.5294**
(LIANG et al., 2025)	0.0300	0.0157

Fonte: o Autor

Como discutido no capítulo 3 de Trabalhos Relacionados, o sistema apresentado por Ebrat et al. (2025) compreende uma combinação das técnicas de filtragem colaborativa, redes de grafos neurais, extração de preferências do usuário-alvo por LLMs e ajuste fino de LLM pela operação de *instruction-tuning*, onde se treina o LLM para aprimorar a inferência dadas instruções esperadas. Já o sistema de (LIANG et al., 2025), trata-se de uma arquitetura que faz uso puramente da capacidade zero-shot do LLM, empregando etapas de extração de contexto sobre itens do usuário-alvo, ao definir estruturas de "taxonomia" representativas da base de dados (gêneros, tema narrativo e idioma), para determinar itens candidatos e realizar a recomendação.

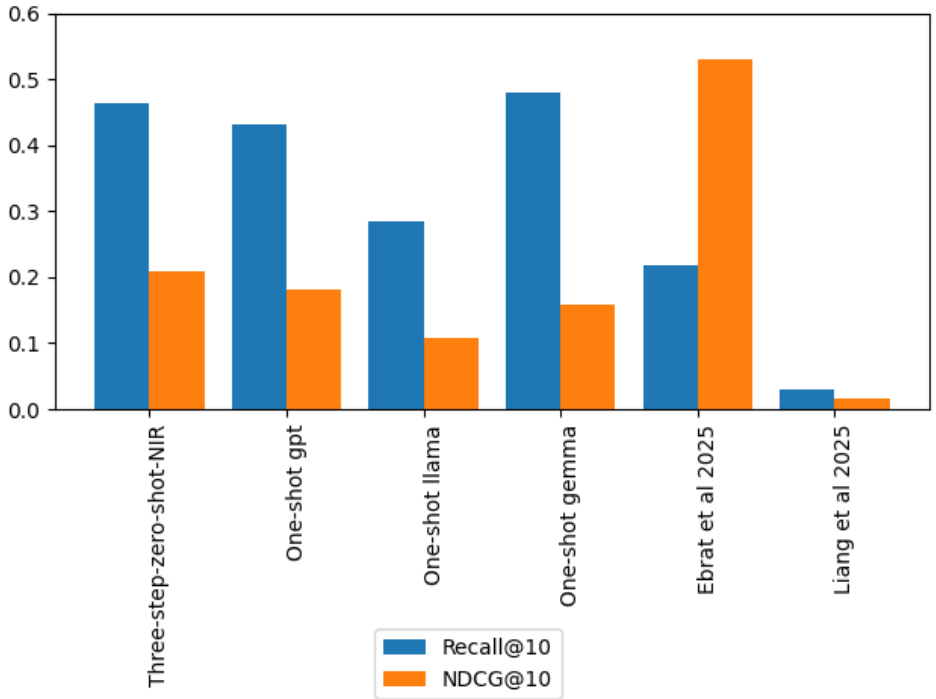
Na perspectiva dada pela métrica *Recall@10*, houve superação dos sistemas externos pelos apresentados neste trabalho. Enfatiza-se, diante de tudo, o observado excelente desempenho do modelo Gemma 3 avaliado, que mostrou grande eficiência nesta métrica, superando tanto os sistemas entre os modelos variantes, quanto os externos. Isso pode se dar pela qualidade de ajuste-fino com *instruction-tuning* característico do modelo (TEAM et al., 2025), bem como a possibilidade de que os dados deste *dataset* tenham passado pelo seu pré-treinamento. Além disso, observa-se o baixíssimo desempenho demonstrado pelo sistema de Liang et al. (2025). Em sua publicação, tendo sido avaliado para a mesma base de dados, pode-se estipular que esse resultado se deu pelo aproveitamento baixo de inferências geradas pelo modelo GPT-4 que foi utilizado, tratando-se de um modelo generalista, enquanto os modelos adotados em todos outros sistemas considerados são de natureza por ajuste fino com a operação de *instruction-tuning*.

Entretanto, em análise da métrica *NDCG@10*, todos sistemas de recomendação apreciados aqui foram superados pelo sistema de Ebrat et al. (2025). Isto significa que os itens *ground-truth* tiveram maior relevância atribuída pelo LLM utilizado, sendo ranqueados para posições mais altas na lista dos top-10 itens recomendados. He et

al. (2015) e Jawansingh e Rai (2025) ressaltam a importância da métrica *NDCG*, dado que ela aponta a qualidade da recomendação sob a perspectiva do nível de utilidade e interesse para o usuário alvo. São identificados ganhos significativos de performance diante dos outros sistemas. Trata-se de mais do que o dobro, quando comparado ao segundo melhor resultado, dado pelo sistema de estratégia *three-step zero-shot*, com ganho de 153.90% entre os valores 0.2085 e 0.5294 de *NDCG@10*. A operação de *instruction-tuning* do LLM utilizado pelo sistema, mostrou-se valoroso para esta performance, dado que a "continuação" do treinamento especificava, especificamente as instruções usadas para recomendação, o que permitiu notável desempenho. Todavia, percebe-se que os sistemas *zero-shot* e *one-shot* abordados se colocaram entre as próximas melhores performances, onde a estratégia *three-step* fez-se superior.

Em suma, com base nos resultados comparados, verificou-se desempenho competitivo dos sistemas *baseline* e dos propostos aqui, ao menos nos aspectos referentes à presença do item ground-truth na recomendação gerada pelo LLM, enquanto o refinado sistema que emprega ajuste fino e redes de grafos neurais, se estabeleceu superior no quesito de qualidade da recomendação, onde itens julgados como favoritos são atribuídos a devida relevância. Para esta base de dados, entre os sistemas dados pelas estratégias que foram objeto de observação deste trabalho, conclui-se que a estratégia original *three-step*, aplicando a técnica zero-shot de engenharia de prompt, gera recomendações de qualidade ligeiramente superior, mediante métrica *NDCG@10*, mesmo que as outras propostas sejam próximas nessa avaliação.

Figura 18 – Comparação de resultados no dataset MovieLens 100K



Fonte: o Autor

5.3.4 Custos computacionais

A utilização da API da OpenAI para computação com o LLM GPT-3.5-Turbo-Instruct envolve um custo compucional contado por *tokens* de processamento. A cobrança pela OpenAI é feita por tipo de token processado (OPENAI, 2026), sendo por token de entrada, os tokens das prompts processados pelo LLM, e os tokens de saída, o texto de geração pelo modelo — cobra-se 1,50$ dólares estadunidenses (USD) por 1 Milhão de tokens de entrada, e 2,00$ dólares por 1 Milhão de tokens de saída, o que significa 0,0000015$ centávos por token de entrada e 0,000002$ centávos por token de saída. Já para a utilização dos modelos Llama e Gemma, não foi necessário nenhum custo por processamento de token, visto que, se tratando de LLMs de acesso gratuito e de código aberto, puderam ser executados num ambiente local no caso dos experimentos feitos neste trabalho.

A tabela 7 mostra a totalização de tokens de entrada e saída para cada estratégia experimentada, que envolveu o envio das prompts formatadas para cada estratégia e a geração das respostas de cada prompt, mostrando também o custo total cobrado pelos tokens processados, em dólares.

Tabela 7 – Custos de utilização do GPT-3.5-Turbo-Instruct

Dataset	**Estratégia**	**Tokens Entrada**	**Tokens Saída**	**Custo($)**
Amazon Beauty	Three-step-prompting	56.670	14.924	0,114
	Two-step-prompting	36.941	13.835	0,083
	Single-step-prompting	16.082	8.086	0,040
	One-shot-prompting	44.287	9.654	0,085
	Chain-of-thoughs-prompting	17.122	10.534	0,046
Amazon Luxury Beauty	Three-step-prompting	1.853.653	499.849	3,780
	Two-step-prompting	1.198.385	457.342	2,712
	Single-step-prompting	520.704	280.588	1,342
	One-shot-prompting	1.402.587	304.215	2,712
	Chain-of-thoughts-prompting	555.908	400.092	1,634
MovieLens 100K	Three-step-prompting	761.941	207.187	1,557
	One-shot-prompting	530.672	116.355	1,028

Fonte: o Autor

Devido ao número pequeno de amostras coletadas para o subconjunto Amazon Beauty, nota-se que a totalização tanto de tokens de entrada quanto de saída foram baixas, vistos que as experimentações nesse caso foram para 22 amostras — dessa forma, recomendações geradas para 22 usuários-alvos ao redor de todas estratégias

de prompt. Em contrapartida, foi processada uma quantidade substâncial de tokens na experimentação com os outros *datasets*, Amazon Luxury Beauty e MovieLens 100K, respectivamente com 830 amostras e 943 amostras.

Um fator acentuado é a diferença da contagem total de tokens entre as duas bases de dados avaliadas, devido à natureza do contexto a que elas se referem. Como o *input* dos itens de recomendação se dá com a extração dos títulos dos produtos a que se referem, nos subsets de análises de produtos Amazon tratam-se de produtos com títulos extensivamente longos que são incluídos nas prompts, enquanto que, no contexto do MovieLens 100K, tem-se a inserção dos títulos dos filmes que, por sua vez, representam sentenças muito curtas, o que torna mais barata a recomendação com os itens desse dataset.

A partir dos números da tabela 7, é possível destacar que consistentemente a estratégia *baseline* teve maior número de tokens de saída entre todas as bases de dados, devido à sua abordagem de três etapas de geração de respostas, o que torna essa a estratégia mais cara no custo total de tokens de saída. A estratégia *two-step-prompting* foi a segunda mais cara em total de tokens *output* por seguir uma ideia semelhante à estratégia baseline, tendo por sua vez duas etapas de geração. Já a estratégia *One-shot-prompting* teve o segundo maior número total de tokens de entrada para todos os datasets, visto que a prompt de entrada dessa abordagem inclui um exemplo de recomendação com os nomes de todos itens candidatos, itens de interação e itens de geração. Isso faz com que a estratégia *one-shot* seja custosa numa escala muito semelhante à da estratégia baseline, mas, ainda assim, mais barata.

5.4 Conclusão

Tomando como base o trabalho de Wang e Lim (2023), as estratégia de prompts foram propostas sob a hipótese da possibilidade de melhoria do desempenho de LLMs em recomendar itens de contextos gerais, vendo que o trabalho definido como *baseline* tinha sido feito sob medida para o contexto da base de dados MovieLens 100K, o que motivou a avaliação em bases de dados diferentes. Indo além, colocou-se a hipótese de que modelos menores, abertos e mais acessíveis poderiam alcançar desempenho semelhantes na tarefa de recomendação, se aplicados estas estratégias propostas.

Analisando os primeiros experimentos no *dataset* Amazon, foi demonstrado que as quatro técnicas de *prompt-engineering* propostas se fazem viáveis em comparação à técnica zero-shot aplicada da forma original, sobretudo no modelo mais robusto GPT-3.5-Turbo-Instruct. O estudo de ablação proposto na seção 5.3.2, destaca a importância do aspecto da engenharia de prompt onde se define o contexto da tarefa a ser executada, percebendo que, a estratégia que apresentou contexto de maior es-

pecificidade e tamanho, e que foi *baseline* da comparação, obteve o segundo melhor desempenho entre as estratégias avaliadas para o *subset* que consistia em 830 amostras. Ainda assim, foi constatado que, para esse mesmo caso que destaca a estratégia *baseline*, quando se incluiu de um exemplo da tarefa como contexto para o LLM, houve a superação ambos em probabilidade de acerto do item *ground-truth*, dado pela métrica HitRate@10 e em qualidade da recomendação, mostrada com a métrica NDCG@10 — isso foi conquistado pela estratégia *one-shot*.

Isso manteve-se consistente ao tornar os experimentos para o *dataset* MovieLens 100K, entretanto, apenas em parte. A estratégia *one-shot* proposta foi superior apenas em taxa de acerto do item relevante *ground-truth* (Recall@10) e, essa performance não se manteve superior no modelo GPT teoricamente mais robusto, tendo o melhor desempenho no modelo Gemma 3, o menor dos modelos. A estratégia *zero-shot* do *baseline* foi, então, superior na métrica NDCG, considerada por He et al. (2015) como um indicador de qualidade da recomendação, conseguindo valor de 0.2085 em NDCG@10 contra 0.1802 em NDCG@10 do melhor resultado da estratégia *one-shot*, representando uma vitória com 15.70% de ganhos.

Analisando sobre as capacidades dos modelos LLM mais acessíveis, percebeu-se que não houve consistência de desempenho numa lógica estratégia-modelo-*dataset*, e de forma geral, ambos modelos Llama 3.1 8B Instruct e Gemma 3-4B-IT tiveram resultados considerados inferiores ou equiparáveis para as duas bases de dados contra o modelo GPT-3.5-Instruct. Apesar disso, ressalta-se que tratam-se de modelos abertos, passíveis de ajuste-fino sem custos por utilização, diferentemente do modelo da OpenAI. Por este motivo e por seus resultados competentes dados seus tamanhos, considera-se estes modelos como viáveis na tarefa de recomendação.

A hipótese de que aplicar técnicas diferentes de engenharia de prompt melhorariam o desempenho da tarefa de recomendação é, portanto, validada em certos aspectos. Faz-se possível essa melhoria de performance, entretanto, apenas parcialmente, visto que aquela julgada como melhor estratégia proposta, foi superada em média NDCG@10 pela estratégia a qual se almejava bater, de forma a concluir que a estratégia *baseline* não é, necessariamente, substituível para o contexto o qual foi originalmente construída. O que se percebe é que há, de fato, a possibilidade de melhoria de desempenho apenas substituindo a estratégia de prompt, mas **qual estratégia de prompt se sairá melhor para cada base de dados** é algo que deve ser investigado aplicando todas as múltiplas possibilidades de técnicas de prompt. Reconhecidamente, devido a restrições de tempo, a abordagem aplicada aqui acabou limitada ao definir apenas uma das estratégias propostas para comparação na execução dos experimentos na base de dados MovieLens, o que, de maneira agora especulativa, poderia demonstrar se outras das estratégias propostas não obteriam resultados melhores no

contexto desse *dataset*.

Enfim, salienta-se que, a conclusão da necessidade de que se investigue qual estratégia de engenharia de prompt se sai melhor em cada contexto de recomendação, ainda avaliando a escolha da combinação estratégia-modelo, se faz especialmente necessária ao imaginar a aplicação de um sistema desse tipo num ambiente produtivo real, seja empresarial, corporativo ou afins, o que seria um contexto onde tal investigação se faz imprescindível.

6 Conclusão

Este trabalho realizou uma análise comparativa dos impactos de um conjunto de diferentes técnicas de engenharia de prompt no desempenho da tarefa de recomendação num sistema baseado em LLMs. Foram propostas quatro estratégias de prompt para recomendação, onde cada uma representava uma técnica de engenharia de prompt específica — nomeadamente, as técnicas *zero-shot* (com duas variantes), *one-shot* e *chain-of-thoughts*, aplicadas em comparação à uma técnica *zero-shot* dada por Wang e Lim (2023), que serviu como *baseline*. Implementou-se um sistema de recomendação que reaproveitou a arquitetura lógica originada por Wang e Lim (2023), constituída de um componente de filtragem colaborativa que define o contexto de preferências dos usuários-alvos, enquanto substituiu-se do componente de solicitação de recomendação, para aplicar as estratégias de prompt que foram objeto desta análise comparativa.

Realizaram-se em duas partes os experimentos, usando em cada parte uma base de dados diferente: Amazon Review Dataset, nos subconjuntos Beauty e Luxury Beauty, e Movie Lens 100K. A partir dos resultados da primeira parte foi possível analisar as causalidades das estratégias de prompts na performance de recomendação entre os três diferentes modelos de grande escala definidos. Sob uma ótica que especificou aspectos para cada estratégia representativa de uma técnica de *prompt-engineering*, demonstrou-se que o fornecimento de exemplos foi o fator determinante para o melhor desempenho na tarefa de recomendação entre os melhores resultados, havendo ganhos de 17.76% na métrica NDCG@10 pela estratégia *one-shot* contra a estratégia *zero-shot*. Entretanto, isso demonstrou-se não completamente consistente na segunda parte dos experimentos, onde se analisou a tarefa de recomendação na base de dados para qual a estratégia do *baseline* foi originalmente construída. Foi analisado a superação na métrica Recall@10 pela estratégia *one-shot* proposta mas, em contraparte, foi evidenciado a superioridade do *baseline* a partir da métrica NDCG@10, que indica a atribuição de relevância dada ao item de teste *ground-truth* pelo LLM, ao contabilizar seu o rank na lista de recomendação gerada. Foi possível concluir também, analisando a partir dos experimentos entre os três modelos avaliados — GPT-3.5-Turbo-Instruct, Llama 3.1 8B Instruct e Gemma-3-4B-IT — a viabilidade que os modelos menos computacionalmente exigentes e, ainda de código aberto, apresentam em oferecer capacidades para recomendação a partir do framework do sistema de recomendação abordado.

Entre todos resultados obtidos **foram evidenciadas melhorias de performance**, entretanto, não houve uma combinação estratégia-modelo que, entre os experimentos

nas duas bases de dados abordadas, obteve desempenho superior a outras combinações ao mesmo tempo para as métricas avaliadas — HitRate@10 e NDCG@10 — de forma que, LLMs diferentes performavam melhor ou pior a depender da estratégia utilizada. Aponta-se, portanto, que a estratégia de prompt utilizada para o sistema de recomendação deve ser escolhida mediante avaliação comparativa entre as estratégias propostas, em combinação com o LLM escolhido, de forma a constatar sua superioridade no contexto da base de dados sobre qual se deseja aplicar a tarefa de recomendação. Ressalta-se ainda mais essa consideração, para o uso imaginado num ambiente produtivo real, visto que esse é o caso onde deve-se avaliar opções mais complexas e "sob medida" para um caso de uso de recomendação com LLMs.

6.1 Limitações

Apesar de apresentar resultados que cumprem a proposta de produção da análise comparativa, existem reconhecidamente algumas limitações na aplicação das melhorias propostas.

Mesmo que a intenção de aplicar a técnica *chain-of-thoughts* tenha sido satisfeita, admite-se que a forma de aplicação dela está entre a mais simples dentre as encontradas na pesquisa de técnicas *COT*. Existe, portanto, uma oportunidade de melhoria na estratégia que aplica esta técnica, que poderia ser implementada sob formas mais refinadas, como aquela apresentada por Zhang et al. (2022), chamada "Auto-COT", buscando melhores resultados para todas as métricas avaliativas.

Também é reconhecida como uma limitação, a aplicação somente da técnica de filtragem por usuários no passo de seleção de itens candidatos, visto que existe a possibilidade de aplicação da outra técnica de Filtragem Colaborativa, chamada Filtragem por Itens. Esta técnica é encontrada na avaliação feita originalmente por Wang e Lim (2023), entretando foi deixada de lado neste trabalho como forma de restrição de escopo avaliativo.

Finalmente, os LLMs *open-source* escolhidos cumprem o papel de representação de modelos acessíveis na análise comparativa, entretanto, é possível julgar que esses são muito pouco robustos para a tarefa de recomendação abordada, pois estão entre as opções mais "fracas" de modelos abertos ainda computacionalmente acessíveis — isto é, passíveis de execução em ambientes de execução como o do Autor, que possui uma GPU não-industrial. Reconhece-se desta forma, que seria possível uma melhoria alavancada pelo uso de ofertas mais potentes entre as versões dos modelos abertos Gemma-3, havendo por exemplo, opções como Gemma-3 27B e, em substituição do Llama 3.1 — cuja versão superior seguinte começa com 70 Bilhões de parâmetros, exigindo maiores recursos computacionais — poderia se utilizar o modelo

gpt-oss-20B da OpenAI, visto estes modelos ainda são de um nível de acessibilidade semelhante aos modelos avaliados neste trabalho.

6.2 Trabalhos Futuros

A partir da pesquisa e das análises produzidas neste trabalho, são considerados os seguintes possíveis trabalhos futuros que sucedem as hipóteses aqui analisadas:

- ***Instruction-Tuning* dos modelos Llama e Gemma para a tarefa de recomendação:** Diante da observação dos resultados obtidos pelo trabalho de Ebrat et al. (2025), pode-se indagar sobre a possibilidade do impacto positivo na qualidade de recomendações geradas, para realizar o ajuste fino dos LLMs acessíveis localmente, com a operação específica de *instruction-tuning*, o que permitiria uma "continuação" da modelagem destes LLMs sob medida para as prompts propostas no contexto específico de uma base de dados definida.
- **Desenvolvimento de técnica *chain-of-thoughts* mais robusta para recomendação:** Desenvolver uma estratégia de prompts para a tarefa de recomendação, onde se aplicasse uma técnica de chain-of-thoughts mais complexa, para buscar resultados robustos de recomendação. A estratégia poderia ser construída a partir da forma "*Auto-CoT*" (ZHANG et al., 2022), para produz *clusters* de exemplos e de perguntas do contexto de recomendação por meio de LLMs, de modo a proceduralmente se formar prompts que serão usadas na tarefa de recomendação.
- **Sistema de Recomendação usando *Retrieval Augmented Generation* com LLM de ajuste fino:** Desenvolver um sistema *RAG* (GAO et al., 2024) que combine um LLM treinado por *instruction-tuning*, juntamente a aplicação de gráfos bipartidos para a preparação dos *chunks* na etapa *Retrieval*, construindo contexto de relacionamento entre usuários e itens. O ajuste fino do LLM escolhido seria dado por um treinamento que adapte o modelo a gerar recomendações, especificamente, com a prompt definida para a última etapa do esquema RAG.

Referências

BAHDANAU, D.; CHO, K.; BENGIO, Y. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. Disponível em: <https://arxiv.org/abs/1409.0473>. Citado na página 22.

BROWN, T. B. et al. *Language Models are Few-Shot Learners*. 2020. Disponível em: <https://arxiv.org/abs/2005.14165>. Citado 4 vezes nas páginas 15, 16, 25 e 26.

CHEN, B. et al. Unleashing the potential of prompt engineering for large language models. *Patterns*, Elsevier BV, v. 6, n. 6, p. 101260, jun. 2025. ISSN 2666-3899. Disponível em: <http://dx.doi.org/10.1016/j.patter.2025.101260>. Citado 3 vezes nas páginas 25, 26 e 27.

CHOI, S.; KIM, W. *Improving the Performance of Sequential Recommendation Systems with an Extended Large Language Model*. 2025. Disponível em: <https://arxiv.org/abs/2507.19990>. Citado 2 vezes nas páginas 33 e 37.

CHOWDHERY, A. et al. *PaLM: Scaling Language Modeling with Pathways*. 2022. Disponível em: <https://arxiv.org/abs/2204.02311>. Citado na página 25.

DEVLIN, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: BURSTEIN, J.; DORAN, C.; SOLORIO, T. (Ed.). *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <https://aclanthology.org/N19-1423/>. Citado 3 vezes nas páginas 20, 23 e 24.

DWICAHYA, I.; ROSA, P. H. P.; NUGROHO, R. A. Movie recommender system comparison of user-based and item-based collaborative filtering systems. In: *Proceedings of the 1st International Conference on Science and Technology for an Internet of Things, 20 October 2018, Yogyakarta, Indonesia*. [S.l.]: EAI, 2019. Citado na página 31.

EBRAT, D. et al. *End-to-End Personalization: Unifying Recommender Systems with Large Language Models*. 2025. Disponível em: <https://arxiv.org/abs/2508.01514>. Citado 4 vezes nas páginas 34, 37, 67 e 75.

GAO, Y. et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. Disponível em: <https://arxiv.org/abs/2312.10997>. Citado na página 75.

Ghaseminejad Raeini, M. The evolution of language models: From n-grams to llms, and beyond. *Natural Language Processing Journal*, v. 12, p. 100168, 2025. ISSN 2949-7191. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2949719125000445>. Citado na página 20.

GRATTAFIORI, A. et al. *The Llama 3 Herd of Models*. 2024. Disponível em: <https://arxiv.org/abs/2407.21783>. Citado na página 47.

GRBOVIC, M. et al. E-commerce in your inbox: Product recommendations at scale. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2015. (KDD '15), p. 1809–1818. ISBN 9781450336642. Disponível em: <https://doi.org/10.1145/2783258.2788627>. Citado 3 vezes nas páginas 14, 27 e 41.

GROUPLENS. *MovieLens 100K Dataset*. 2026. Acessado em 21 de janeiro de 2026. Disponível em: <https://grouplens.org/datasets/movielens/100k/>. Citado na página 54.

HAMILTON, W.; YING, Z.; LESKOVEC, J. Inductive representation learning on large graphs. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf>. Citado na página 14.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 5, n. 4, dez. 2015. ISSN 2160-6455. Disponível em: <https://doi.org/10.1145/2827872>. Citado 3 vezes nas páginas 31, 41 e 49.

HE, X. et al. Trirank: Review-aware explainable recommendation by modeling aspects. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2015. (CIKM '15), p. 1661–1670. ISBN 9781450337946. Disponível em: <https://doi.org/10.1145/2806416.2806504>. Citado 3 vezes nas páginas 50, 68 e 71.

HE, X. et al. Lightgcn: Simplifying and powering graph convolution network for recommendation. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2020. (SIGIR '20), p. 639–648. ISBN 9781450380164. Disponível em: <https://doi.org/10.1145/3397271.3401063>. Citado na página 35.

HE, X. et al. Neural collaborative filtering. In: *Proceedings of the 26th International Conference on World Wide Web*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017. (WWW '17), p. 173–182. ISBN 9781450349130. Disponível em: <https://doi.org/10.1145/3038912.3052569>. Citado 2 vezes nas páginas 50 e 54.

HIRSCHBERG, J.; MANNING, C. D. Advances in natural language processing. *Science*, v. 349, n. 6245, p. 261–266, 2015. Disponível em: <https://www.science.org/doi/abs/10.1126/science.aaa8685>. Citado 2 vezes nas páginas 14 e 20.

JAWANSINGH, R. S.; RAI, A. K. Evaluating recommender system using baseline approaches. *Procedia Computer Science*, v. 258, p. 2323–2333, 2025. ISSN 1877-0509. International Conference on Machine Learning and Data Engineering. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050925015984>. Citado na página 68.

JIANG, Z. et al. Instruction-tuned language models are better knowledge learners. In: KU, L.-W.; MARTINS, A.; SRIKUMAR, V. (Ed.). *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Bangkok, Thailand: Association for Computational Linguistics, 2024. p. 5421–5434. Disponível em: <https://aclanthology.org/2024.acl-long.296/>. Citado na página 47.

JURAFSKY, D.; MARTIN, J. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2009. (Pearson international edition). ISBN 9780135041963. Disponível em: <https://books.google.com.br/books?id=crxYPgAACAAJ>. Citado na página 20.

KANG, W.-C.; MCAULEY, J. *Self-Attentive Sequential Recommendation*. 2018. Disponível em: <https://arxiv.org/abs/1808.09781>. Citado 4 vezes nas páginas 30, 36, 37 e 51.

KOJIMA, T. et al. *Large Language Models are Zero-Shot Reasoners*. 2023. Disponível em: <https://arxiv.org/abs/2205.11916>. Citado na página 45.

LI, Y. et al. *Recent Developments in Recommender Systems: A Survey*. 2023. Disponível em: <https://arxiv.org/abs/2306.12680>. Citado na página 27.

LIANG, Y. et al. Taxonomy-guided zero-shot recommendations with LLMs. In: RAMBOW, O. et al. (Ed.). *Proceedings of the 31st International Conference on Computational Linguistics*. Abu Dhabi, UAE: Association for Computational Linguistics, 2025. p. 1520–1530. Disponível em: <https://aclanthology.org/2025.coling-main.102/>. Citado 3 vezes nas páginas 32, 37 e 67.

MIKOLOV, T. et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. Disponível em: <https://arxiv.org/abs/1301.3781>. Citado na página 20.

MIN, S. et al. *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* 2022. Disponível em: <https://arxiv.org/abs/2202.12837>. Citado na página 16.

MINAEE, S. et al. *Large Language Models: A Survey*. 2025. Disponível em: <https://arxiv.org/abs/2402.06196>. Citado 3 vezes nas páginas 22, 23 e 25.

NAVEED, H. et al. A comprehensive overview of large language models. *ACM Trans. Intell. Syst. Technol.*, Association for Computing Machinery, New York, NY, USA, v. 16, n. 5, ago. 2025. ISSN 2157-6904. Disponível em: <https://doi.org/10.1145/3744746>. Citado 2 vezes nas páginas 22 e 26.

NAWARA, D.; KASHEF, R. A comprehensive survey on llm-powered recommender systems: From discriminative, generative to multi-modal paradigms. *IEEE Access*, v. 13, p. 145772–145798, 2025. Citado 2 vezes nas páginas 50 e 51.

NI, J.; LI, J.; MCAULEY, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: INUI, K. et al. (Ed.). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 188–197. Disponível em: <https://aclanthology.org/D19-1018/>. Citado 2 vezes nas páginas 41 e 49.

OLLAMA. *Ollama*. 2026. Acessado em 30 de janeiro de 2026. Disponível em: <https://github.com/ollama/ollama?tab=readme-ov-file>. Citado na página 49.

OPENAI. *GPT-4 API general availability and deprecation of older models in the Completions API*. 2024. Acessado em 21 de janeiro de 2026. Disponível em: <https://openai.com/index/gpt-4-api-general-availability>. Citado na página 47.

OPENAI. *OpenAI Pricing*. 2026. Acessado em 01 de fevereiro de 2026. Disponível em: <https://platform.openai.com/docs/pricing?legacy-pricing=standard>. Citado 2 vezes nas páginas 17 e 69.

PENG, Q. et al. A survey on LLM-powered agents for recommender systems. In: CHRISTODOULOPOULOS, C. et al. (Ed.). *Findings of the Association for Computational Linguistics: EMNLP 2025*. Suzhou, China: Association for Computational Linguistics, 2025. p. 11574–11583. ISBN 979-8-89176-335-7. Disponível em: <https://aclanthology.org/2025.findings-emnlp.620/>. Citado 5 vezes nas páginas 14, 15, 29, 33 e 49.

RADFORD, A. et al. *Improving language understanding by generative pre-training*. 2018. Disponível em: <https://openai.com/index/language-unsupervised/>. Citado 3 vezes nas páginas 14, 24 e 28.

RAZA, S. et al. A comprehensive review of recommender systems: Transitioning from theory to practice. *Computer Science Review*, v. 59, p. 100849, 2026. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S157401372500125X>. Citado 5 vezes nas páginas 14, 15, 27, 28 e 29.

RENDLE, S. et al. *BPR: Bayesian Personalized Ranking from Implicit Feedback*. 2012. Disponível em: <https://arxiv.org/abs/1205.2618>. Citado na página 35.

SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th International Conference on World Wide Web*. New York, NY, USA: Association for Computing Machinery, 2001. (WWW '01), p. 285–295. ISBN 1581133480. Disponível em: <https://doi.org/10.1145/371920.372071>. Citado na página 31.

SCHICK, T.; SCHüTZE, H. *Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference*. 2021. Disponível em: <https://arxiv.org/abs/2001.07676>. Citado na página 20.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, v. 61, p. 85–117, 2015. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>. Citado na página 21.

SHENBIN, I. et al. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2020. (WSDM '20), p. 528–536. ISBN 9781450368223. Disponível em: <https://doi.org/10.1145/3336191.3371831>. Citado na página 14.

SRILAKSHMI, M.; CHOWDHURY, G.; SARKAR, S. Two-stage system using item features for next-item recommendation. *Intelligent Systems with Applications*, v. 14, p. 200070, 2022. ISSN 2667-3053. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667305322000114>. Citado na página 41.

TEAM, G. et al. *Gemma 3 Technical Report*. 2025. Disponível em: <https://arxiv.org/abs/2503.19786>. Citado 2 vezes nas páginas 47 e 67.

TOUVRON, H. et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2302.13971>. Citado na página 25.

TURING, A. M. Computing machinery and intelligence. *Mind*, [Oxford University Press, Mind Association], v. 59, n. 236, p. 433–460, 1950. ISSN 00264423, 14602113. Disponível em: <http://www.jstor.org/stable/2251299>. Citado na página 20.

VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>. Citado 2 vezes nas páginas 21 e 22.

WANG, L.; LIM, E.-P. *Zero-Shot Next-Item Recommendation using Large Pretrained Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2304.03153>. Citado 21 vezes nas páginas 9, 15, 18, 31, 36, 37, 38, 39, 42, 44, 47, 49, 54, 57, 60, 64, 66, 67, 70, 73 e 74.

WANG, T. et al. *What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?* 2022. Disponível em: <https://arxiv.org/abs/2204.05832>. Citado na página 23.

WANG, X. et al. Neural graph collaborative filtering. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2019. (SIGIR'19), p. 165–174. ISBN 9781450361729. Disponível em: <https://doi.org/10.1145/3331184.3331267>. Citado na página 35.

WU, L. et al. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 2022. ISSN 2326-3865. Disponível em: <http://dx.doi.org/10.1109/TKDE.2022.3145690>. Citado na página 27.

WU, L. et al. *A Survey on Large Language Models for Recommendation*. 2024. Disponível em: <https://arxiv.org/abs/2305.19860>. Citado na página 49.

XIE, X. et al. *Contrastive Learning for Sequential Recommendation*. 2021. Disponível em: <https://arxiv.org/abs/2010.14395>. Citado na página 57.

YU, S. et al. Predicting the next sentence (not word) in large language models: What model-brain alignment tells us about discourse comprehension. *Science Advances*, v. 10, n. 21, p. eadn7744, 2024. Disponível em: <https://www.science.org/doi/abs/10.1126/sciadv.adn7744>. Citado na página 22.

YUE, Z. et al. *LlamaRec: Two-Stage Recommendation using Large Language Models for Ranking*. 2023. Disponível em: <https://arxiv.org/abs/2311.02089>. Citado na página 33.

YUE, Z. et al. Linear recurrent units for sequential recommendation. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2024. (WSDM '24), p. 930–938. ISBN 9798400703713. Disponível em: <https://doi.org/10.1145/3616855.3635760>. Citado na página 33.

ZHANG, J. et al. *AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems*. 2023. Disponível em: <https://arxiv.org/abs/2310.09233>. Citado na página 29.

ZHANG, S. et al. Instruction tuning for large language models: A survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 58, n. 7, jan. 2026. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3777411>. Citado na página 47.

ZHANG, Z. et al. *Automatic Chain of Thought Prompting in Large Language Models*. 2022. Disponível em: <https://arxiv.org/abs/2210.03493>. Citado 2 vezes nas páginas 74 e 75.

ÇANO, E.; MORISIO, M. Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, SAGE Publications, v. 21, n. 6, p. 1487–1524, nov. 2017. ISSN 1571-4128. Disponível em: <http://dx.doi.org/10.3233/IDA-163209>. Citado na página 28.

A Apêndice

A.1 Prompts de entrada das estratégias aplicadas para recomendação

Encontra-se nessa seção os exemplos das prompts formatadas com itens de usuários-alvos e itens candidatos de cada estratégia de prompt avaliada.

A.1.1 Prompts das estratégias *Zero-shot*

Snippet A.1 – Prompt de recomendação da estratégia *baseline Three-step-prompting*

```
Candidate Set (candidate products of all beauty category): Kingfansion Nail
    Art Stamping Stamper Scraper Image Plate Transfer Manicure Tool, Born
   Pretty Nail Art Water Decals Transfer Sticker 2 Patterns/Sheet Flower
   Leaves, BORN PRETTY Nail Art Stamp Template image stamping plates Cute
   Snow Heart Pattern QA86, Susenstone Christmas DIY Image Stamp Stamping
   Plates Manicure Template Nail Art Plate, Nail Art Stickers Sandistore 12
   pcs Flower Water Transfer Women Manicure Nail Art Stickers DIY Tips
   Decoration (#4), Urban Spa Natural Bamboo and Jute Bath Mitt, Neutrogena
    Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl. oz., Goody
   Simple Styles Spin Pin Dark Hair, Generic 12Pcs Nail Art Water Decals
   Transfer Stickers Chic Pink Floral Pattern C8-001, BMC Nail Stamping
   Lacquers Creative Art Polish Collection 6 Colors: Set 1, Oral-B Glide
   Pro-Health Dental Floss Original Floss 50m Pack of 6, Colgate Fluoride
   Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack of 6), Kingfansion
   Peel Off Liquid Tape Latex Tape Peel Off Base Coat Nail Art Liquid
   Palisade (Blue ) by kingfansion, BORN PRETTY 1 Sheet Nail Wraps
   Mysterious Starry Sky Night Patterned Full Nail Sticker, Panasonic
   Bikini Shaper and Trimmer for Women ES246AC; Compact; Portable Design
   with Adjustable Trim Settings; Battery Operated, Nail Art Stickers
   Sandistore S Nail Art Image Stamp Stamping Plates Manicure Template Hehe
    Series (#1), Born Pretty Nail Art Stamping Template Image Plate Chic
   Rose Flower BP65, Clean & Clear Deep Action Cream Facial Cleanser for
   Sensitive Skin Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4),
   Born Pretty Flower Owl Nail Art Water Decals Set Transfer Sticker 10
   Sheets/Set #20698, Malloom 1pc Butterfly Nail Art Sticker Stamping
   Stainless Steel Plates DIY Decoration.
The products I have reviewed (reviewed of all beauty category): Super Nail
   Polish Thinner 4 Ounce (118ml), Qimisi 12 Color Glitter Hexagon Acrylic
   UV Gel False Tips Nail Art Salon Tool Set, Kingfansion Nail Stamping
   Printing Plate Image Stamps Plate Manicure Nail Art Decor, Kingfansion
   XL Silicone Dual Ended Nail Stamp Scraper Stamper Transfer Stamping
```

Plate, BMC 4pc DIY Decal Making Nail Stamping Metal Guide Templates, Kingfansion Nail Stamping Printing Plate Image Stamps Plate Nail Art Decor Manicure, DANCINGNAIL New 17m White Stripe Tape Roll Nail Art Manicure Edge Guides Tips Tool DIY Sticker Line 0.5cm Decoration, Dragonpad 10pcs Round Nail Art Display for Practice Wheel Arylic Tips Tool, Bundle Monster 10pc Holiday Themed Nail Art Stamping Plates Occasions Collection Halloween + Thanksgiving.

Step 1: What features are most important to me when selecting a product (Summarize my preferences briefly)?

Answer: I prefer products that are easy to use, have a variety of designs or colors, and are affordable. I also value products that are durable and have good reviews from other customers..

Step 2: Selecting the most featured products from the reviewed products according to my preferences (Format: [no. a reviewed product.]).

Answer: 1. Kingfansion Nail Art Stamping Stamper Scraper Image Plate Transfer Manicure Tool

2. Born Pretty Nail Art Water Decals Transfer Sticker 2 Patterns/Sheet Flower Leaves
3. BORN PRETTY Nail Art Stamp Template image stamping plates Cute Snow Heart Pattern QA86
4. Susenstone Christmas DIY Image Stamp Stamping Plates Manicure Template Nail Art Plate
5. Nail Art Stickers Sandistore 12pcs Flower Water Transfer Women Manicure Nail Art Stickers DIY Tips Decoration (#4)
6. Urban Spa Natural Bamboo and Jute Bath Mitt
7. Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl. oz.
8. Goody Simple Styles Spin Pin Dark Hair
9. Generic 12Pcs Nail Art Water Decals Transfer Stickers Chic Pink Floral Pattern C8-001
10. BMC Nail Stamping Lacquers Creative Art Polish Collection 6 Colors: Set 1
11. Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
12. Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack of 6)
13. Kingfansion Peel Off Liquid Tape Latex Tape Peel Off Base Coat Nail Art Liquid Palisade (Blue) by kingfansion
14. BORN PRETTY 1 Sheet Nail Wraps Mysterious Starry Sky Night Patterned Full Nail Sticker
15. Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact; Portable Design with Adjustable Trim Settings; Battery Operated
16. Nail Art Stickers Sandistore S Nail Art Image Stamp Stamping Plates Manicure Template Hehe Series (#1)
17. Born Pretty Nail Art Stamping Template Image Plate Chic Rose Flower BP65
18. Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)

```
19. Born Pretty Flower Owl Nail Art Water Decals Set Transfer Sticker 10
    Sheets/Set #20698
20. Malloom 1pc Butterfly Nail Art Sticker Stamping Stainless Steel Plates
    DIY Decoration..
Step 3: Can you recommend 10 products from the Candidate Set similar to the
     selected products I've reviewed (Format: [no. a reviewed product - a
    candidate product])?.
Answer:
```

Snippet A.2 – Prompt 2 de recomendação da estratégia *Two-step-prompting*

```
Candidate Set (candidate products of all beauty category): Detangling
    Hairbrush  Gentle Detangler Hair Brush & Comb  No More Tangle  Reduce
    Hair Loss and Breakage  Great for Adults & Kids  Lime Green, BeautyMe
    Blackhead and Pimple Remover Kit with 7 Surgical Extractor Tool, Crest +
     Oral-B Professional Gingivitis Kit 1 Count, Crest Sensi-Stop Strips 10
    Count, NARS Blush Taj Mahal, 9 Pcs Manicure Set Pedicure Tools Nail Care
     Art Kit For Girls Teens Women Men Includes Nail Files Emery Boards Nail
     Clipper Cuticle Sticks Manicure Gloves Travel Pouch by Perfect Life
    Ideas, Crest Pro-health Multi-Protection Rinse Cool Wintergreen 33.8
    Fluid Ounce, Urban Spa Natural Bamboo and Jute Bath Mitt, Svelta Luxe
    Coffee & Raw Sugar Body Scrub / Reduce Appearance of Cellulite and
    Stretch Marks / Long Lasting Hydration / Deep Intense Exfoliation /
    Invigorating with Natural Oils and Organic Sugar, Urban Spa Moisturizing
     Booties to Keep your Feet Smooth Hydrated and Moisturized, Pantene Pro-
    V Volume Conditioner 12.0 Fluid Ounce (Product Size May Vary), Crest Pro
    -Health For Life CPC Antigingivitis/Antiplaque Smooth Mint Rinse 33.8 Fl
     Oz, Dove Men+Care Deep Clean Body + Face Bar 4 Ounce 6 Count (Pack of
    2), Crest + Oral-B Professional Daily Clean Kit 1 Count, essie Gel
    Couture Nail Polish, Nadira Organics Virgin Argan Oil for Skin Face Hair
     and Nails 4 fl. oz., Beard Trimmer Kit 5 in 1 Multi-functional Body
    Groomer Kit of Mustache Trimmer Nose Hair Trimmer and Precision Trimmer
    Waterproof and Rechargeable Cordless (BT114S), Renova Red Toilet Paper
    -6 pack, Piero Lorenzo NT-01 Heavy Duty Metal Garden Hose Nozzle Sprayer
     / Car Wash Gun- 7 Spraying Patterns High Pressure For Car / Pet Washing
     Garden/Lawn WateringDeck/Floor Cleaning7 Spraying Patterns, Hotrose
    Shower Bath Back Brush Scrubber Skin Cleaning Body Massager.
The products I have reviewed (reviewed of all beauty category): Panasonic
    Bikini Shaper and Trimmer for Women ES246AC; Compact; Portable Design
    with Adjustable Trim Settings; Battery Operated, Clean & Clear Deep
    Action Cream Facial Cleanser for Sensitive Skin Gentle Daily Face Wash
    with Oil-Free 6.5 oz (Pack of 4), Colgate Fluoride Toothpaste Strawberry
     Smash Liquid Gel 4.60 oz (Pack of 6), Oral-B Glide Pro-Health Dental
    Floss Original Floss 50m Pack of 6, Goody Simple Styles Spin Pin Dark
    Hair, Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3
     Fl. oz., Vktech Hair Braider Twist Styling Braid Tool Magic Wonder
    Holder Clip DIY French, Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz
    6.660-Fluid Ounce, Wavertree & London Lavender D'Provence (8 bars) -
```

```
    Triple-milled (twice) Shea Butter soap Bar -Rich & Creamy Lather, Pre de
     Provence Artisanal French Soap Bar Enriched with Shea Butter Quad-
    Milled For A Smooth & Rich Lather (150 grams) Raspberry, Colgate Kids
    Maximum Cavity Protection Pump Toothpaste 4.4 ounce (12 Pack), Suave for
     Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml), Coppertone Water
    Babies SPF Sunblock Stick .6oz, Simply Beautiful De Tangle Brush
    Professional Detangling Hairbrush Pink Black Purple Blue or Green (Black
    ), Beauty Bridge Anti-Aging Protective Day Cream, Wavertree & London
    Beach (8 bars) -Triple-milled (twice) Shea Butter soap Bar -Rich &
    Creamy Lather, Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster
     Brush set for Hair Cut Hairstylist Design Gown Barbers Salon Make up
    Cosmetic.
Given my Candidate Set
Given my most relevant products:  1. Panasonic Bikini Shaper and Trimmer
    for Women ES246AC; Compact; Portable Design with Adjustable Trim
    Settings; Battery Operated
2. Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin
    Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
3. Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack of
     6)
4. Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
5. Goody Simple Styles Spin Pin Dark Hair
6. Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl.
    oz.
7. Vktech Hair Braider Twist Styling Braid Tool Magic Wonder Holder Clip
    DIY French
8. Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz 6.660-Fluid Ounce
9. Wavertree & London Lavender D'Provence (8 bars) -Triple-milled (twice)
    Shea Butter soap Bar -Rich & Creamy Lather
10. Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
    Quad-Milled For A Smooth & Rich Lather (150 grams) Raspberry
Recommend 10 products from the Candidate Set similar to my most relevant
    products (Format: [no. a relevant product - a candidate product])
Answer:
```

Snippet A.3 – Prompt de recomendação da estratégia *Single-step-prompting*

```
Candidate Set (candidate products of all beauty category): Piero Lorenzo NT
    -01 Heavy Duty Metal Garden Hose Nozzle Sprayer / Car Wash Gun- 7
    Spraying Patterns High Pressure For Car / Pet Washing Garden/Lawn
    WateringDeck/Floor Cleaning7 Spraying Patterns, Detangling Hairbrush
    Gentle Detangler Hair Brush & Comb  No More Tangle  Reduce Hair Loss and
     Breakage  Great for Adults & Kids  Lime Green, Urban Spa Natural Bamboo
     and Jute Bath Mitt, Crest Pro-health Multi-Protection Rinse Cool
    Wintergreen 33.8 Fluid Ounce, Nadira Organics Virgin Argan Oil for Skin
    Face Hair and Nails 4 fl. oz., Crest Sensi-Stop Strips 10 Count, Hotrose
     Shower Bath Back Brush Scrubber Skin Cleaning Body Massager, NARS Blush
     Taj Mahal, Pantene Pro-V Volume Conditioner 12.0 Fluid Ounce (Product
```

```
    Size May Vary), Svelta Luxe Coffee & Raw Sugar Body Scrub / Reduce
    Appearance of Cellulite and Stretch Marks / Long Lasting Hydration /
    Deep Intense Exfoliation / Invigorating with Natural Oils and Organic
    Sugar, 9 Pcs Manicure Set Pedicure Tools Nail Care Art Kit For Girls
    Teens Women Men Includes Nail Files Emery Boards Nail Clipper Cuticle
    Sticks Manicure Gloves Travel Pouch by Perfect Life Ideas, Beard Trimmer
     Kit 5 in 1 Multi-functional Body Groomer Kit of Mustache Trimmer Nose
    Hair Trimmer and Precision Trimmer Waterproof and Rechargeable Cordless
    (BT114S), BeautyMe Blackhead and Pimple Remover Kit with 7 Surgical
    Extractor Tool, Crest + Oral-B Professional Gingivitis Kit 1 Count,
    Renova Red Toilet Paper -6 pack, Dove Men+Care Deep Clean Body + Face
    Bar 4 Ounce 6 Count (Pack of 2), Urban Spa Moisturizing Booties to Keep
    your Feet Smooth Hydrated and Moisturized, Crest Pro-Health For Life CPC
     Antigingivitis/Antiplaque Smooth Mint Rinse 33.8 Fl Oz, essie Gel
    Couture Nail Polish, Crest + Oral-B Professional Daily Clean Kit 1 Count
    .
The products I have reviewed (reviewed of all beauty category): Panasonic
    Bikini Shaper and Trimmer for Women ES246AC; Compact; Portable Design
    with Adjustable Trim Settings; Battery Operated, Clean & Clear Deep
    Action Cream Facial Cleanser for Sensitive Skin Gentle Daily Face Wash
    with Oil-Free 6.5 oz (Pack of 4), Colgate Fluoride Toothpaste Strawberry
     Smash Liquid Gel 4.60 oz (Pack of 6), Oral-B Glide Pro-Health Dental
    Floss Original Floss 50m Pack of 6, Goody Simple Styles Spin Pin Dark
    Hair, Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3
     Fl. oz., Vktech Hair Braider Twist Styling Braid Tool Magic Wonder
    Holder Clip DIY French, Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz
    6.660-Fluid Ounce, Wavertree & London Lavender D'Provence (8 bars) -
    Triple-milled (twice) Shea Butter soap Bar -Rich & Creamy Lather, Pre de
     Provence Artisanal French Soap Bar Enriched with Shea Butter Quad-
    Milled For A Smooth & Rich Lather (150 grams) Raspberry, Colgate Kids
    Maximum Cavity Protection Pump Toothpaste 4.4 ounce (12 Pack), Suave for
     Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml), Coppertone Water
    Babies SPF Sunblock Stick .6oz, Simply Beautiful De Tangle Brush
    Professional Detangling Hairbrush Pink Black Purple Blue or Green (Black
    ), Beauty Bridge Anti-Aging Protective Day Cream, Wavertree & London
    Beach (8 bars) -Triple-milled (twice) Shea Butter soap Bar -Rich &
    Creamy Lather, Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster
     Brush set for Hair Cut Hairstylist Design Gown Barbers Salon Make up
    Cosmetic.
Recommend 10 products from the Candidate Set similar to 10 of the most
    relevant products for me from the products I have reviewed (Format: [no.
     a relevant product - a candidate product])
Answer:
```

A.1.2 Prompt da estratégia *Chain-of-Thoughts*

Snippet A.4 – Prompt de recomendação da estratégia *Chain-of-thoughts-prompting*

```
The user chooses products in the all beauty category to consume based on
    its preferences according to previously reviewed products.
The user has a set of previously reviewed products
(
     Super Nail Polish Thinner 4 Ounce (118ml)
     Qimisi 12 Color Glitter Hexagon Acrylic UV Gel False Tips Nail Art
    Salon Tool Set
     Kingfansion Nail Stamping Printing Plate Image Stamps Plate Manicure
    Nail Art Decor
     Kingfansion XL Silicone Dual Ended Nail Stamp Scraper Stamper Transfer
    Stamping Plate
     BMC 4pc DIY Decal Making Nail Stamping Metal Guide Templates
     Kingfansion Nail Stamping Printing Plate Image Stamps Plate Nail Art
    Decor Manicure
     DANCINGNAIL New 17m White Stripe Tape Roll Nail Art Manicure Edge
    Guides Tips Tool DIY Sticker Line 0.5cm Decoration
     Dragonpad 10pcs Round Nail Art Display for Practice Wheel Arylic Tips
    Tool
     Bundle Monster 10pc Holiday Themed Nail Art Stamping Plates Occasions
    Collection Halloween + Thanksgiving
)
and a set of candidate products
(
     Generic 12Pcs Nail Art Water Decals Transfer Stickers Chic Pink Floral
    Pattern C8-001
     Born Pretty Nail Art Water Decals Transfer Sticker 2 Patterns/Sheet
    Flower Leaves
     Nail Art Stickers Sandistore 12pcs Flower Water Transfer Women Manicure
     Nail Art Stickers DIY Tips Decoration (#4)
     Malloom 1pc Butterfly Nail Art Sticker Stamping Stainless Steel Plates
    DIY Decoration
     BORN PRETTY Nail Art Stamp Template image stamping plates Cute Snow
    Heart Pattern QA86
     Born Pretty Flower Owl Nail Art Water Decals Set Transfer Sticker 10
    Sheets/Set #20698
     BMC Nail Stamping Lacquers Creative Art Polish Collection 6 Colors: Set
     1
     BORN PRETTY 1 Sheet Nail Wraps Mysterious Starry Sky Night Patterned
    Full Nail Sticker
     Kingfansion Nail Art Stamping Stamper Scraper Image Plate Transfer
    Manicure Tool
     Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
     Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl.
    oz.
     Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin
    Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
```

```
    Nail Art Stickers Sandistore S Nail Art Image Stamp Stamping Plates
   Manicure Template Hehe Series (#1)
    Goody Simple Styles Spin Pin Dark Hair
    Kingfansion Peel Off Liquid Tape Latex Tape Peel Off Base Coat Nail Art
    Liquid Palisade (Blue ) by kingfansion
    Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack
   of 6)
    Born Pretty Nail Art Stamping Template Image Plate Chic Rose Flower
   BP65
    Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact;
   Portable Design with Adjustable Trim Settings; Battery Operated
    Susenstone Christmas DIY Image Stamp Stamping Plates Manicure Template
   Nail Art Plate
    Urban Spa Natural Bamboo and Jute Bath Mitt
).
What would be a selection of 10 recommended product for the user, based on
   its preferences, following the format [no. a relevant product - a
   candidate product]?
Let's think step by step.
```

A.1.3 Prompts das estratégias *One-shot*

Snippet A.5 – Prompt *one-shot* formatada com uma amostra do subconjunto *Beauty* da base de dados *Amazon Reviews Dataset* usando o exemplo 1 de recomendação definido

```
A user with candidate set of all beauty products
(
    Wavertree & London Beach (8 bars) -Triple-milled (twice) Shea Butter
   soap Bar -Rich & Creamy Lather
    Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
    Suave for Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml)
    Pre de Provence Maison French Dried Lavender Blossoms for Fragrance
    Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster Brush set for
    Hair Cut Hairstylist Design Gown Barbers Salon Make up Cosmetic
    Wavertree & London Lavender D'Provence (8 bars) -Triple-milled (twice)
   Shea Butter soap Bar -Rich & Creamy Lather
    Urban Spa Moisturizing Booties to Keep your Feet Smooth Hydrated and
   Moisturized
    Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin
   Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
    Coppertone Water Babies SPF Sunblock Stick .6oz
    Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact;
   Portable Design with Adjustable Trim Settings; Battery Operated
    Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl.
   oz.
```

```
    Beauty Bridge Anti-Aging Protective Day Cream
    Urban Spa Natural Bamboo and Jute Bath Mitt
    Vktech Hair Braider Twist Styling Braid Tool Magic Wonder Holder Clip
   DIY French
    Revitalash By Revitalash Revitalash Advanced Eyelash Conditioner-- 3.5
   Ml / .118 Oz
    Colgate Enamel Health Mouthwash
    Simply Beautiful De Tangle Brush Professional Detangling Hairbrush Pink
     Black Purple Blue or Green (Black)
    Goody Simple Styles Spin Pin Dark Hair
    MAKE UP FOR EVER Mist & Fix 4.22 oz
    Pre De Provence Maison French Lavender Bath & Shower Gel
)
and a set of previously reviewed products
(
    Crest Pro-health Multi-Protection Rinse Cool Wintergreen 33.8 Fluid
   Ounce
    Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack
   of 6)
    Crest + Oral-B Professional Gingivitis Kit 1 Count
    Crest Pro-Health For Life CPC Antigingivitis/Antiplaque Smooth Mint
   Rinse 33.8 Fl Oz
    Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz 6.660-Fluid Ounce
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (150 grams) Raspberry
    Hotrose Shower Bath Back Brush Scrubber Skin Cleaning Body Massager
    Dove Men+Care Deep Clean Body + Face Bar 4 Ounce 6 Count (Pack of 2)
    Colgate Kids Maximum Cavity Protection Pump Toothpaste 4.4 ounce (12
   Pack)
    BeautyMe Blackhead and Pimple Remover Kit with 7 Surgical Extractor
   Tool
    Beard Trimmer Kit 5 in 1 Multi-functional Body Groomer Kit of Mustache
   Trimmer Nose Hair Trimmer and Precision Trimmer Waterproof and
   Rechargeable Cordless (BT114S)
    Crest + Oral-B Professional Daily Clean Kit 1 Count
    Pantene Pro-V Volume Conditioner 12.0 Fluid Ounce (Product Size May
   Vary)
    9 Pcs Manicure Set Pedicure Tools Nail Care Art Kit For Girls Teens
   Women Men Includes Nail Files Emery Boards Nail Clipper Cuticle Sticks
   Manicure Gloves Travel Pouch by Perfect Life Ideas
)
receives as recommendation the following 10 products from the candidate set
    of products, based on his preferences on his previosly reviewed
   products, formatted as [no. a relevant product - a candidate product]:

(
```

1. Crest Pro-health Multi-Protection Rinse Cool Wintergreen 33.8 Fluid Ounce - Urban Spa Moisturizing Booties to Keep your Feet Smooth Hydrated and Moisturized
2. Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack of 6) - Colgate Enamel Health Mouthwash
3. Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter Quad -Milled For A Smooth & Rich Lather (150 grams) Raspberry - Wavertree & London Lavender D'Provence (8 bars) -Triple-milled (twice) Shea Butter soap Bar -Rich & Creamy Lather
4. Dove Men+Care Deep Clean Body + Face Bar 4 Ounce 6 Count (Pack of 2) - Suave for Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml)
5. BeautyMe Blackhead and Pimple Remover Kit with 7 Surgical Extractor Tool - Vktech Hair Braider Twist Styling Braid Tool Magic Wonder Holder Clip DIY French
6. Beard Trimmer Kit 5 in 1 Multi-functional Body Groomer Kit of Mustache Trimmer Nose Hair Trimmer and Precision Trimmer Waterproof and Rechargeable Cordless (BT114S) - Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact; Portable Design with Adjustable Trim Settings; Battery Operated
7. Pantene Pro-V Volume Conditioner 12.0 Fluid Ounce (Product Size May Vary) - Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
8. 9 Pcs Manicure Set Pedicure Tools Nail Care Art Kit For Girls Teens Women Men Includes Nail Files Emery Boards Nail Clipper Cuticle Sticks Manicure Gloves Travel Pouch by Perfect Life Ideas - Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster Brush set for Hair Cut Hairstylist Design Gown Barbers Salon Make up Cosmetic
9. Crest + Oral-B Professional Gingivitis Kit 1 Count - Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
10. Crest Pro-Health For Life CPC Antigingivitis/Antiplaque Smooth Mint Rinse 33.8 Fl Oz - MAKE UP FOR EVER Mist & Fix 4.22 oz

)

Another user with candidate set of all beauty products

(

Wavertree & London Beach (8 bars) -Triple-milled (twice) Shea Butter soap Bar -Rich & Creamy Lather

Urban Spa Natural Bamboo and Jute Bath Mitt

Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6

Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)

Simply Beautiful De Tangle Brush Professional Detangling Hairbrush Pink Black Purple Blue or Green (Black)

Beauty Bridge Anti-Aging Protective Day Cream

Paul Brown Hawaii Hapuna Hair Styling Paste 8 Ounce

Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl. oz.

```
    Crest + Oral-B Professional Gingivitis Kit 1 Count
    Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz 6.660-Fluid Ounce
    Suave for Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml)
    Vktech Hair Braider Twist Styling Braid Tool Magic Wonder Holder Clip
   DIY French
    Crest + Oral-B Professional Daily Clean Kit 1 Count
    Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster Brush set for
    Hair Cut Hairstylist Design Gown Barbers Salon Make up Cosmetic
    Goody Simple Styles Spin Pin Dark Hair
    Wavertree & London Lavender D'Provence (8 bars) -Triple-milled (twice)
   Shea Butter soap Bar -Rich & Creamy Lather
    Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact;
   Portable Design with Adjustable Trim Settings; Battery Operated
    Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack
   of 6)
    Colgate Kids Maximum Cavity Protection Pump Toothpaste 4.4 ounce (12
   Pack)
    Coppertone Water Babies SPF Sunblock Stick .6oz
)
and a set of previously reviewed products (
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (250 grams) Milk
    Michel Design Works Bath Soap Bar Peony Large
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (150 grams) Raspberry
    Bundle of 2 Greenwich Bay Trading Co. Soaps 10.5oz Bath Soap Bar and
   Matching 1.9oz Hand Soap Bar (Exfoliating Pomegranate Shea Butter)
    Michel Design Works Oversized Triple Milled Bath Soap Bar Avocado Large
    8.7 Ounce
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (250 grams) Rose Petal
    Calgon Massaging Beauty Bar English Garden: 2 Bars
    Vinolia Cold Cream Bath Soap 170g (1)
    CST Box Bath Soap (Cherry Blossom)
    Asquith & Somerset Triple Milled Luxury Soap Bar Love Tea Rose
    La Vie En Rose Natural Bar Soap Rose Geranium Essential Oil 7 Oz
    Asquith and Somerset Luxury Poinsettia Bath Soap 10.5oz
    Christmas Scent Holiday Soap Frosted Mint 12 Oz BAR
)
receives as recommendation the following 10 products from the candidate set
    of products, based on his preferences on his previosly reviewed
   products, formatted as [no. a relevant product - a candidate product]:
```

Snippet A.6 – Prompt *one-shot* formatada com uma amostra do subconjunto *Beauty* da base de dados *Amazon Reviews Dataset* usando o exemplo 2 de recomendação definido

```
A user with candidate set of all beauty products
(
    Wavertree & London Beach (8 bars) -Triple-milled (twice) Shea Butter
   soap Bar -Rich & Creamy Lather
    Urban Spa Natural Bamboo and Jute Bath Mitt
    Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
    Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin
   Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
    Simply Beautiful De Tangle Brush Professional Detangling Hairbrush
   Pink Black Purple Blue or Green (Black)
    Beauty Bridge Anti-Aging Protective Day Cream
    Paul Brown Hawaii Hapuna Hair Styling Paste 8 Ounce
    Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl.
    oz.
    Crest + Oral-B Professional Gingivitis Kit 1 Count
    Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz 6.660-Fluid Ounce
    Suave for Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml)
    Vktech Hair Braider Twist Styling Braid Tool Magic Wonder Holder Clip
   DIY French
    Crest + Oral-B Professional Daily Clean Kit 1 Count
    Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster Brush set
   for Hair Cut Hairstylist Design Gown Barbers Salon Make up Cosmetic
    Goody Simple Styles Spin Pin Dark Hair
    Wavertree & London Lavender D'Provence (8 bars) -Triple-milled (twice)
    Shea Butter soap Bar -Rich & Creamy Lather
    Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact;
   Portable Design with Adjustable Trim Settings; Battery Operated
    Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack
   of 6)
    Colgate Kids Maximum Cavity Protection Pump Toothpaste 4.4 ounce (12
   Pack)
    Coppertone Water Babies SPF Sunblock Stick .6oz
)
and a set of previously reviewed products
(
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (250 grams) Milk
    Michel Design Works Bath Soap Bar Peony Large
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (150 grams) Raspberry
    Bundle of 2 Greenwich Bay Trading Co. Soaps 10.5oz Bath Soap Bar and
   Matching 1.9oz Hand Soap Bar (Exfoliating Pomegranate Shea Butter)
    Michel Design Works Oversized Triple Milled Bath Soap Bar Avocado Large
    8.7 Ounce
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (250 grams) Rose Petal
    Calgon Massaging Beauty Bar English Garden: 2 Bars
```

```
    Vinolia Cold Cream Bath Soap 170g (1)
    CST Box Bath Soap (Cherry Blossom)
    Asquith & Somerset Triple Milled Luxury Soap Bar Love Tea Rose
    La Vie En Rose Natural Bar Soap Rose Geranium Essential Oil 7 Oz
    Asquith and Somerset Luxury Poinsettia Bath Soap 10.5oz
    Christmas Scent Holiday Soap Frosted Mint 12 Oz BAR
)
receives as recommendation the following 10 products from the candidate set
     of products, based on his preferences on his previosly reviewed
    products, formatted as [no. a relevant product - a candidate product]:

(
1. Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter Quad
    -Milled For A Smooth & Rich Lather (250 grams) Milk - Paul Brown Hawaii
    Hapuna Hair Styling Paste 8 Ounce
2. Michel Design Works Bath Soap Bar Peony Large - Urban Spa Natural Bamboo
     and Jute Bath Mitt
3. Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter Quad
    -Milled For A Smooth & Rich Lather (150 grams) Raspberry - Fekkai Full
    Blown Aerosol Foam Cond Us 6.6 Oz 6.660-Fluid Ounce
4. Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter Quad
    -Milled For A Smooth & Rich Lather (250 grams) Rose Petal - Wavertree &
    London Lavender D'Provence (8 bars) -Triple-milled (twice) Shea Butter
    soap Bar -Rich & Creamy Lather
5. Vinolia Cold Cream Bath Soap 170g (1) - Wavertree & London Beach (8 bars
    ) -Triple-milled (twice) Shea Butter soap Bar -Rich & Creamy Lather
6. Asquith & Somerset Triple Milled Luxury Soap Bar Love Tea Rose - Aisilk
    Hair Cutting Cape Hairdressing & Large Neck Duster Brush set for Hair
    Cut Hairstylist Design Gown Barbers Salon Make up Cosmetic
7. La Vie En Rose Natural Bar Soap Rose Geranium Essential Oil 7 Oz -
    Beauty Bridge Anti-Aging Protective Day Cream
8. Asquith and Somerset Luxury Poinsettia Bath Soap 10.5oz - Vktech Hair
    Braider Twist Styling Braid Tool Magic Wonder Holder Clip DIY French
9. Christmas Scent Holiday Soap Frosted Mint 12 Oz BAR - Neutrogena Ultra
    Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl. oz.
10. Bundle of 2 Greenwich Bay Trading Co. Soaps 10.5oz Bath Soap Bar and
    Matching 1.9oz Hand Soap Bar (Exfoliating Pomegranate Shea Butter) -
    Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin
    Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
)

Another user with candidate set of all beauty products
(
    Panasonic Bikini Shaper and Trimmer for Women ES246AC; Compact;
    Portable Design with Adjustable Trim Settings; Battery Operated
    MAKE UP FOR EVER Mist & Fix 4.22 oz
    Pre De Provence Maison French Lavender Bath & Shower Gel
```

```
    Beauty Bridge Anti-Aging Protective Day Cream
    Wavertree & London Beach (8 bars) -Triple-milled (twice) Shea Butter
   soap Bar -Rich & Creamy Lather
    Clean & Clear Deep Action Cream Facial Cleanser for Sensitive Skin
   Gentle Daily Face Wash with Oil-Free 6.5 oz (Pack of 4)
    Goody Simple Styles Spin Pin Dark Hair
    Pre de Provence Maison French Dried Lavender Blossoms for Fragrance
    Neutrogena Ultra Sheer Dry-Touch Sunscreen Broad Spectrum SPF 70 3 Fl.
   oz.
    Coppertone Water Babies SPF Sunblock Stick .6oz
    Suave for Kids 2 in 1 Shampoo Dragon Fruit 12 fl oz (355 ml)
    Aisilk Hair Cutting Cape Hairdressing & Large Neck Duster Brush set for
    Hair Cut Hairstylist Design Gown Barbers Salon Make up Cosmetic
    Colgate Enamel Health Mouthwash
    Revitalash By Revitalash Revitalash Advanced Eyelash Conditioner-- 3.5
   Ml / .118 Oz
    Simply Beautiful De Tangle Brush Professional Detangling Hairbrush Pink
    Black Purple Blue or Green (Black)
    Urban Spa Natural Bamboo and Jute Bath Mitt
    Oral-B Glide Pro-Health Dental Floss Original Floss 50m Pack of 6
    Vktech Hair Braider Twist Styling Braid Tool Magic Wonder Holder Clip
   DIY French
    Wavertree & London Lavender D'Provence (8 bars) -Triple-milled (twice)
   Shea Butter soap Bar -Rich & Creamy Lather
    Urban Spa Moisturizing Booties to Keep your Feet Smooth Hydrated and
   Moisturized
)
and a set of previously reviewed products (
    Crest Pro-health Multi-Protection Rinse Cool Wintergreen 33.8 Fluid
   Ounce
    Colgate Fluoride Toothpaste Strawberry Smash Liquid Gel 4.60 oz (Pack
   of 6)
    Crest + Oral-B Professional Gingivitis Kit 1 Count
    Crest Pro-Health For Life CPC Antigingivitis/Antiplaque Smooth Mint
   Rinse 33.8 Fl Oz
    Fekkai Full Blown Aerosol Foam Cond Us 6.6 Oz 6.660-Fluid Ounce
    Pre de Provence Artisanal French Soap Bar Enriched with Shea Butter
   Quad-Milled For A Smooth & Rich Lather (150 grams) Raspberry
    Hotrose Shower Bath Back Brush Scrubber Skin Cleaning Body Massager
    Dove Men+Care Deep Clean Body + Face Bar 4 Ounce 6 Count (Pack of 2)
    Colgate Kids Maximum Cavity Protection Pump Toothpaste 4.4 ounce (12
   Pack)
    BeautyMe Blackhead and Pimple Remover Kit with 7 Surgical Extractor
   Tool
    Beard Trimmer Kit 5 in 1 Multi-functional Body Groomer Kit of Mustache
   Trimmer Nose Hair Trimmer and Precision Trimmer Waterproof and
   Rechargeable Cordless (BT114S)
```

```
    Crest + Oral-B Professional Daily Clean Kit 1 Count
    Pantene Pro-V Volume Conditioner 12.0 Fluid Ounce (Product Size May
    Vary)
    9 Pcs Manicure Set Pedicure Tools Nail Care Art Kit For Girls Teens
    Women Men Includes Nail Files Emery Boards Nail Clipper Cuticle Sticks
    Manicure Gloves Travel Pouch by Perfect Life Ideas
)
receives as recommendation the following 10 products from the candidate set
    of products, based on his preferences on his previosly reviewed
    products, formatted as [no. a relevant product - a candidate product]:
```

Snippet A.7 – Prompt *one-shot* formatada com uma amostra da base de dados MovieLens 100K e com o exemplo de recomendação definido

```
A user with candidate set of movies
(
    Under Siege 2: Dark Territory
    Natural Born Killers
    The Three Musketeers
    Batman Returns
    GoldenEye
    Batman Forever
    Days of Thunder
    Star Trek III: The Search for Spock
    Full Metal Jacket
    Star Trek VI: The Undiscovered Country
    The Crow
    Speed
    Money Train
    Stargate
    True Lies
    Conan the Barbarian
    Under Siege
    First Knight
    Batman
)
and a set of previously reviewed movies
(
    The Maltese Falcon
    Con Air
    Romy and Michele's High School Reunion
    Anastasia
    Grosse Pointe Blank
    The Fifth Element
    Starship Troopers
    Wild America.
)
```

```
receives as recommendation the following 10 movies from the candidate set
   of movies, based on his preferences on his previosly reviewed movies,
   formatted as [no. a relevant movie - a candidate movie]:

(
    1. The Fifth Element - Star Trek III: The Search for Spock
    2. Starship Troopers - Star Trek VI: The Undiscovered Country
    3. Con Air - Speed
    4. Grosse Pointe Blank - True Lies
    5. Wild America - Days of Thunder
    6. The Fifth Element - Batman Returns
    7. Starship Troopers - GoldenEye
    8. Con Air - Under Siege
    9. Grosse Pointe Blank - The Crow
    10. Wild America - First Knight
)

Another user with candidate set of movies
(
    Batman
    Liar Liar
    Scream
    Stand by Me
    Executive Decision
    Schindler's List
    Mission: Impossible
    Heathers
    Clueless
    Grease
    That Thing You Do!
    Heat
    The Aristocats
    Speed
    Interview with the Vampire
    Mrs. Doubtfire
    The Crying Game
    E.T. the Extra-Terrestrial
    True Lies
)
and a set of previously reviewed movies
(
    Three Colors: White
    A Grand Day Out
    Desperado
    Glengarry Glen Ross
    Angels and Insects
    Groundhog Day
```

```
    Delicatessen
    The Hunt for Red October
    Dirty Dancing
    The Rock
    Ed Wood
    Star Trek: First Contact
)
receives as recommendation the following 10 movies from the candidate set
   of movies, based on his preferences on his previosly reviewed movies,
   formatted as [no. a relevant movie - a candidate movie]:
```

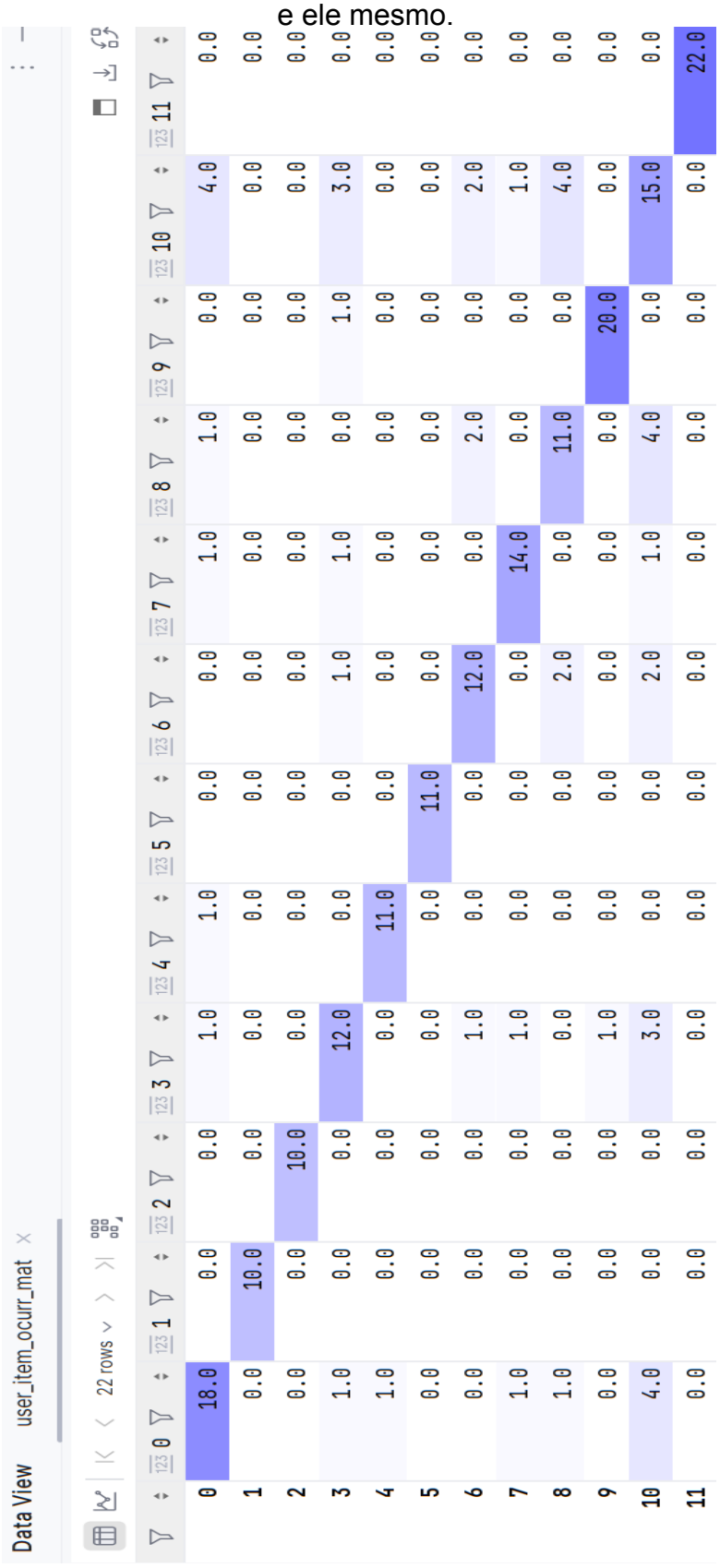
A.2 Exemplos de artefatos da filtragem por usuários

Tabela 8 – Exemplo de 15 maiores somatórios dos pesos de itens candidatos para um usuário-alvo

Id do Produto	**Valor peso-similaridade**
B002GP80EU	0.6896551724137931
B00CZH3K1C	0.24137931034482757
B000NKJIXM	0.1724137931034483
B0010ZBORW	0.1724137931034483
B00DY59MB6	0.1724137931034483
B00CZH3LQG	0.1724137931034483
B013G464EM	0.13793103448275862
B016V8YWBC	0.13793103448275862
B01CHS3CHA	0.13793103448275862
B01G08QMAW	0.13793103448275862
B00N2WQ2IW	0.13793103448275862
B011ABK2LO	0.13793103448275862
B00L1I1VMG	0.06896551724137931
B00OXDWFG2	0.06896551724137931
B00UVZ58GA	0.06896551724137931

Figura 19 – Matriz *user-user* de coocorrências de interações em itens entre usuários

A imagem mostra a contagem de coocorrência de mesmos itens interagidos entre cada par de usuários, nesse caso, cada linha e cada coluna sendo respectiva a um usuário. É perceptível que, cada par de índices iguais em linha e coluna, terá o maior somatório, pois trata-se da comparação de mesmos itens interagidos entre o usuário e ele mesmo.



A.3 Snippets

Snippet A.8 – Snippet código de pré-processamento dos dados brutos da base dados Amazon Reviews Dataset

```
import os
import gzip
import json
import numpy as np
import datetime
import argparse
import html

from pathlib import Path

def parse(path):
'''
Função de carregamento de dados do dataset
'''
    g = gzip.open(path, 'rt', encoding='utf-8') # Carrega arquivo .json para memória
    for l in g:
        yield json.loads(l) # retorna cada linha do arquivo como um dicionário python, atributos e valores no formato ('chave': valor)

def parse_single_quotations(path):
'''
Função de carregamento de dados do dataset, para tramento do arquivo .json em chaves duplas
'''
    g = gzip.open(path, 'r') # Carrega arquivo .json para memória
    for l in g:
        yield eval(l) # retorna cada linha do arquivo como um dicionário python, atributos e valores no formato ('chave': valor)

''' Necessário remover reviews de produtos que não estejam no metadata, pois não terão titulo e impossibilita a criação da prompt '''

metadata_dict = {}  # Dicionário para carregamento dos metadados dos itens, chave 'asin' é o id do produto
for metadata in tqdm(parse_single_quotations(metadata_path), total=metadata_len, desc="Loading metadata"):  # Iterador criado para pré-processamento dos metadados
    if 'title' in metadata: # Carrega o item apenas se tiver atributo de título do produto, a ser usado na recomendação
        metadata_dict[metadata['asin']] = metadata  # Carrega metadados do item para o dicionário declarado
        metadata_dict[metadata['asin']]['title'] = html.unescape(metadata['title']) # Título dos produtos vêm com 'sujos' com caracteres de markdown HTML, a função html.unescape converte o markdown para a string respectiva

def load_data():
'''
Função criada definição das coleções de registros do dataset
'''
    review_list = []    # A lista de todas análises válidas do dataset
    product_subset_dict = {}    # dicionário de todos produtos, mapeando chave de id
```

```
    do produto para seus os metadados
41
42      for review in tqdm(parse(reviews_path), total=reviews_len, desc="Loading reviews
        "):
43          if review['asin'] in metadata_dict: # Barra registros que tiverem de Id do
        produto não está mapeada entre os itens que contém título
44              review_list.append(review)  # Carrega cada análise para a lista de
        análises
45
46              if 'asin' in review and review['asin'] not in product_subset_dict:
47                  product_subset_dict[review['asin']] = metadata_dict[review['asin']]
        # Cria nova entrada no dicionário de produtos
48
49      user_reviews_dict = {}  # Dicionário de usuários com suas respectivas análises
50      for review in review_list:    # Para cada análise
51          if review['reviewerID'] not in user_reviews_dict:
52              user_reviews_dict[review['reviewerID']] = {}
53          user_reviews_dict[review['reviewerID']][review['asin']] = review    #
        Adiciona registro de análise no dicionário de análises do usuário
54
55      return review_list, product_subset_dict, user_reviews_dict # Retorna uma lista
        de análises, um dicionário de todos produtos e um dicionário de usuários para
        suas análises
```

Snippet A.9 – Contabilização das métricas *HitRate@10* e *NDCG@10*

```
1  '''
2  Definição das métricas avaliativas HitRate@10 e NDCG@10.
3  Dada um conjunto de recomendações geradas para uma amostra de uma base de dados, com
       uma estratégia de prompts escolhida, se contabiliza ambos HitRate@10 e NDCG@10
       para cada instância de recomendação, avaliando, respectivamente, a presença do
       item ground-truth na recomendação e a posição em que ele se encontra na lista
       gerada.
4  Uma instância se refere a recomendação gerada a partir do template formatado da
       prompt que contém conjunto de itens de interação do usuário-alvo e itens
       candidatos.
5  Ao fim de todas iterações, terá calculado a média das métricas HitRate@10 e NDCG@10
       entre todas as recomendações geradas.
6  '''
7
8  results_dict, processed_data_filename = load_results() #    Carrega o arquivo que
       contém as recomendações geradas
9  user_candidate_items = load_user_candidate_items(processed_data_filename)
10 recom_key = get_strat_recom_key()   # Carrega a referência da prompt de recomendação
       da estratégia, devido a casos de prompts múltiplas
11 eval_dict = {'results_exec_log_filename': results_filename}
12 header_keys = ['processed_dataset_filename', 'input_token_count', 'model', 'comment'
       ]
13
14 hit = 0
15 ndcg_at10 = 0
16 total_count = 0
17
18 for user_id_key, log_dict in results_dict.items():  # Para cada amostra de usuário-
       alvo
19
```

```
20      if user_id_key not in header_keys:  # Ignora atributos definidos como cabeçalho
    contendo informações de log
21
22          user_sample = user_candidate_items[user_id_key]
23          candidate_items = user_sample['canddts']        # Carrega itens candidatos
    do usuário-alvo
24          user_ground_truth = user_sample['tst_canddt']   # Carrega o item ground-
    truth do usuário-alvo
25
26          llm_recom = log_dict[recom_key] # Carrega recomendação gerada
27
28          total_count += 1    # Contador de amostras avaliadas
29          instance_ndcg = 0
30
31          if user_ground_truth in llm_recom:  #   Ambos HitRate e NDCG são
    contabilizados na presença do item ground-truth na recomendação
32              hit += 1    #   Conta o acerto
33              rank = rank_ground_truth(llm_recom, candidate_items, user_ground_truth)
    # Ranqueia a posição do ground-truth na lista
34
35              if 0 < rank <= 10:
36                  instance_ndcg = 1 / np.log2(1 + rank)   # Cálculo do NDCG@10 desta
    instância
37
38          ndcg_at10 += instance_ndcg
39
40          instance_eval = {}
41          instance_eval['hit'] = hit
42          instance_eval['hit@10'] = hit/total_count   # Define a média HitRate@10
    entre as amostras até então
43          instance_eval['ndcg'] = instance_ndcg
44          instance_eval['ndcg@10_total'] = ndcg_at10
45          instance_eval['ndcg_at10'] = ndcg_at10/total_count  # Define a média NDCG@10
     entre as amostras até então
46          instance_eval['total_count'] = total_count
47
48          '''
49          A ultima amostra avaliada contabilizará a média HitRate@10 e NDCG@10 entre
    todas amostras de recomendação
50          '''
51          eval_dict[user_id_key] = instance_eval
```