Calibração de Sistemas de Recomendação com LLMs: Otimização de *Prompts* para Balancear Precisão, Diversidade e Justiça

Gabriel Prenassi[1], Rodrigo Souza[2], Henrique Sekido[2], Guilherme Fonseca[3], Marcelo G. Manzato[2], Leonardo Rocha[1]
[1]Universidade Federal de São João del-Rei, [2]Universidade de São Paulo, [3]Universidade Federal de Minas Gerais
prenassigabriel@aluno.ufsj.edu.br;(rodrigofsouza,riqueyseki04)@usp.br;guilhermefonseca@dcc.ufmg.br
mmanzato@icmc.usp.br;lcrocha@ufsj.edu.br

ABSTRACT

Recommender Systems (RSs) play a central role in digital platforms, aiming to deliver relevant and personalized content. However, issues like popularity bias persist, limiting diversity and fairness. Calibration techniques seek to align recommendations with user preferences, often through post-processing adjustments. With the advent of Large Language Models (LLMs) such as GPT and LLaMA, new opportunities have emerged to personalize recommendations using prompt engineering. While recent approaches like prompt optimization have shown improvements in ranking accuracy, they often neglect other key aspects such as diversity, coverage, and fairness. This study investigates the use of LLMs for recommendation calibration, comparing their performance against traditional methods. We also evaluate the impact of different prompt optimization strategies across multiple metrics, including MAP, NDCG@10, MRMC, LTC, and GAP. Additionally, we employ a multicriteria utility function (MAUT) to analyze trade-offs between accuracy and diversity. Our results highlight the potential of LLMs and prompt engineering to enhance both personalization and fairness in RSs.

KEYWORDS

Recomendação, LLM, Engenharia de *Prompt*, Viés de Popularidade

1 INTRODUÇÃO

Sistemas de Recomendação (SsR) estão inseridos no cotidiano das pessoas em diversas plataformas de mídia digital, auxiliando na compra de produtos, na sugestão de locais relevantes [8, 40] e na descoberta de músicas [7, 9, 10], sempre com o objetivo de fornecer conteúdo relevante e personalizado [28, 31, 41]. No entanto, apesar dos avanços significativos alcançados com técnicas de aprendizado de máquina e inteligência artificial, desafios importantes ainda persistem, como o viés de popularidade, que favorece itens populares em detrimento dos menos conhecidos, limitando a diversidade das recomendações [23]. Tais desafios são abordados na literatura por diferentes perspectivas, incluindo estratégias interativas para reduzir a dependência inicial em itens populares [33]. Esses problemas impactam negativamente não apenas a satisfação dos usuários, mas também a descoberta de conhecimento, um dos objetivos de SsR [22].

Uma abordagem promissora para mitigar tais problemas é a calibração das recomendações [32, 35, 36], que visa alinhar os resultados sugeridos pelo sistema com as preferências expressas ou implícitas do usuário. Em outras palavras, um sistema calibrado busca refletir de maneira proporcional o histórico de consumo do indivíduo, evitando, por exemplo, que apenas um gênero ou categoria domine as recomendações. Tradicionalmente, essa técnica é aplicada como uma etapa de pós-processamento, por meio de reponderações, interpolação de *rankings* ou ajustes nos escores [3, 12, 32].

Recentemente, os avanços em arquiteturas neurais profundas, notadamente os Grandes Modelos de Linguagem (*Large Language Models* – LLMs), como GPT e LLaMA, vêm abrindo novas possibilidades para a calibração de SsR [27, 37]. Tais modelos possibilitam o uso de engenharia de *prompts* e cadeias de pensamento (*chain-of-thought*) para uma reclassificação personalizada e calibrada [17], bem como otimização em linguagem natural para controlar e personalizar recomendações [27]. Alguns desses trabalhos mostram que é possível otimizar *prompts* para a reclassificação de recomendações, obtendo ganhos entre 5,6% e 20,7% em métricas como NDCG@10 [39]. Algo comum a todos esses trabalhos é que concentram-se principalmente em melhorar a precisão do *ranking*, negligenciando aspectos importantes como diversidade, cobertura, justiça e viés de popularidade. Essa tendência também é destacada por Bittencourt et al. [5], que apontam que muitos estudos em SsR priorizam métricas de precisão em detrimento de outros critérios, como diversidade e justiça.

Este trabalho tem como objetivo geral investigar o potencial dos LLMs na calibração de sistemas de recomendação, com foco na promoção de recomendações mais balanceadas entre precisão, diversidade e justiça. Embora existam estudos comparando o desempenho de abordagens tradicionais de calibração com versões baseadas em LLMs [27], as análises são limitadas por não explorarem diferentes estratégias de *prompting*. Para validação da abordagem proposta, optamos pelo cenário de filmes, clássico em SsR [4, 18]. Nesse sentido, propomos uma abordagem em duas etapas: primeiro, comparamos diretamente modelos tradicionais e LLMs, neste trabalho utilizamos o LLama3.1 como modelo de base, na tarefa de recomendação/calibração; em seguida, avaliamos o impacto da otimização de *prompts* sobre diferentes métricas de avaliação, considerando múltiplas estratégias de refinamento. A proposta visa não apenas melhorar a acurácia das recomendações, mas também mitigar desigualdades induzidas por vieses, como o de popularidade, promovendo experiências de recomendação mais equitativas e personalizadas. Assim, o trabalho busca responder às seguintes perguntas de pesquisa:

RQ1: *O uso de estratégias baseadas em LLMs proporciona ganhos sobre métodos tradicionais de recomendação em relação a métricas como precisão, diversidade, cobertura e justiça?*

RQ2: *O quanto a otimização de prompts de estratégias baseadas em LLMs pode aprimorar a qualidade das recomendações em termos de precisão, diversidade, cobertura e justiça?*

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2025). Rio de Janeiro, Brazil. Porto Alegre: Brazilian Computer Society, 2025.

ISSN 2966-2753

Para responder a essas perguntas, comparamos diferentes estratégias de otimização de *prompts*, avaliando seu impacto em múltiplas métricas importantes no domínio de recomendação, como MAP, NDCG@10, MRMC (*Mean Rank Miscalibration*) de Gêneros, MRMC de Popularidade, LTC (*Long Tail Coverage*) e GAP (*Group Average Popularity*). Adicionalmente, propomos o uso de uma métrica agregada baseada na Teoria da Utilidade MultiAtributo (MAUT) [6, 26], permitindo analisar os *trade-offs* existentes entre precisão, popularidade e diversidade de forma integrada. Os resultados são contrastados com abordagens clássicas de recomendação, como BPR (*Bayesian Personalized Ranking*) [29], calibração por gêneros [36] e calibração por popularidade [2]. Em relação à **RQ1**, nossos resultados apontam que SsR baseados em LLMs superam as abordagens tradicionais em precisão, garantindo também um melhor equilíbrio entre as métricas de diversidade/cobertura. Em relação à **RQ2**, os resultados indicam que a otimização de *prompts* permite calibrar os objetivos do sistema de recomendação, mas não oferece ganhos universais. Enquanto alguns modelos preservam a precisão com custo em equidade, outros ampliam a diversidade de exposição com leve perda em desempenho. Assim, a escolha da estratégia de otimização deve considerar as prioridades específicas de cada aplicação.

Este artigo está estruturado da seguinte forma: na Seção 2, apresentamos uma análise dos trabalhos relacionados. Na Seção 3, descrevemos nossa proposta de recomendação baseada em LLMs, bem como os processos de otimização. Na Seção 4, detalhamos como os experimentos foram configurados. A Seção 5 apresenta os resultados obtidos e discussões sobre eles. Por fim, na Seção 6 são apresentadas as conclusões e direcionamentos para trabalhos futuros.

2 TRABALHOS RELACIONADOS

O viés de popularidade e a calibração de SsR são temas amplamente discutidos na literatura, dada a sua relevância para melhorar a diversidade, justiça e satisfação do usuário. Diversas abordagens vêm sendo propostas para mitigar o impacto desse viés, buscando equilibrar a exposição de itens populares e de nicho [34]. Um trabalho seminal nesse campo é o de Steck [36], que propôs um método de pós-processamento que ajusta a distribuição dos itens recomendados para alinhar-se às preferências históricas dos usuários, particularmente ajustando as proporções de gêneros de itens consumidos. Essa abordagem pioneira estabeleceu a base para diversas pesquisas que visam calibrar recomendações conforme o perfil do usuário.

De forma complementar, Abdollahpouri et al. [2] classificam os usuários em perfis distintos de preferência quanto à popularidade dos itens (nicho, diverso e *blockbuster*) e desenvolvem sistemas que adaptam as recomendações para refletir essas preferências. Essa personalização permite uma avaliação mais apurada do impacto da popularidade nas experiências dos usuários, promovendo uma melhor adequação entre as recomendações e expectativas individuais.

Outras pesquisas avançam na mitigação do viés por meio da integração de modelos preditivos com técnicas de pós-processamento. Sacilotti et al. [32] propuseram uma estratégia que combina essas etapas, reduzindo o viés de popularidade enquanto mantém a relevância dos itens recomendados. Já De et al. [14] levaram essa ideia para o estágio de treinamento, modificando o algoritmo BPR [29] para incorporar diretamente a calibração das preferências de popularidade no processo de aprendizado. Embora essas abordagens tenham demonstrado eficácia, elas dependem majoritariamente de ajustes nos escores ou reclassificação das recomendações, não explorando as vantagens trazidas pela compreensão semântica e flexibilidade dos Grandes Modelos de Linguagem (LLMs).

Nesse contexto, os LLMs abriram novas possibilidades para sistemas de recomendação. A engenharia de *prompts* surge como um mecanismo poderoso para guiar esses modelos, permitindo a adaptação e refinamento das recomendações de forma dinâmica e personalizada. Liu et al. [25] apresentam o *RecPrompt*, uma arquitetura de *prompts* autoajustável para recomendação de notícias, que otimiza iterativamente os *prompts* e alcança ganhos significativos. Revisões recentes, como a de Wu et al. [42], oferecem uma visão abrangente dos paradigmas da recomendação generativa, destacando quatro elementos essenciais na engenharia de *prompts*: a descrição da tarefa, a modelagem dos interesses do usuário, a construção dos candidatos e as estratégias específicas de *prompting*, as quais conferem aos sistemas maior flexibilidade e adaptabilidade.

Além da capacidade técnica, estudos recentes apontam para o potencial dos LLMs na mitigação do viés de popularidade. Ortega et al. [27] analisaram recomendações *zero-shot* produzidas pelo ChatGPT-3.5 Turbo no domínio de filmes, mostrando que as mesmas apresentam menor viés de popularidade e maior justiça do que métodos tradicionais. Lichtenberg et al. [24] corroboram esses resultados, evidenciando que recomendadores *zero-shot* baseados em GPT-3.5 superam modelos colaborativos tradicionais em termos de viés, atribuível ao conhecimento semântico intrínseco dos LLMs. Zhang et al. [45] complementam esse quadro ao avaliar a imparcialidade do ChatGPT em diversos domínios, utilizando o *benchmark* FaiRLLM, que revela desequilíbrios na exposição de itens populares, ressaltando a necessidade de métodos controlados de calibração. Zhao et al. [47], por sua vez, focaram diretamente na calibração de *prompts* para recomendação, demonstrando que ajustes adequados nesses *prompts* melhoram a relevância e a personalização dos resultados.

Apesar dos avanços, nossa análise da literatura revela uma lacuna significativa: poucos estudos exploram de forma sistemática e integrada como diferentes estratégias de otimização de *prompts* impactam simultaneamente a precisão, diversidade, cobertura e justiça das recomendações produzidas por LLMs. Muitas pesquisas focam em um único critério, geralmente a precisão, deixando de lado aspectos fundamentais para a qualidade e equidade do sistema, além da escassez de trabalhos que comparem diretamente modelos tradicionais com abordagens baseadas em LLMs sob múltiplas métricas e diferentes cenários de otimização de *prompts*. Assim, o presente trabalho propõe-se a preencher essa lacuna, avaliando de forma abrangente o impacto dessas estratégias em sistemas de recomendação calibrados com LLMs, adotando uma abordagem multicritério e comparando com técnicas tradicionais consolidadas.

3 RECOMENDAÇÃO BASEADA EM LLM

Em nossa proposta de recomendação com LLM, adotamos o *Llama3.1-8b-instruct*, quantizado em 4 *bits*, o que reduz significativamente o consumo de memória da GPU sem causar prejuízos relevantes no desempenho [20]. Optamos por esse modelo por ser aberto e amplamente utilizado na literatura [11, 15, 16]. O procedimento adotado consistiu na solicitação de recomendações personalizadas por meio do *prompt* apresentado na Figura 1, instanciado para uma coleção de dados relacionada a filmes. Esse *prompt* é composto por duas partes: o `System Prompt`, que fornece o contexto ao modelo; e o `User`

`Prompt`, que representa a entrada fornecida ao modelo, composta por uma solicitação de recomendação personalizada construída a partir dos 20 itens previamente consumidos pelo usuário, extraídos do conjunto de treino. As saídas geradas pelo modelo foram limitadas a até 1000 novos *tokens*, com amostragem ativada e configuração de *temperature* para 0.7 e *top-p* em 0.9. O parâmetro *use_cache* foi mantido ativado para otimizar o desempenho da inferência.

***Prompt* Inicial**

System Prompt:
Given movies/tv shows titles, provide the recommendations following pattern:
1. title (release year)
2. title (release year)
...
User Prompt:
I need exactly 10 movies or TV shows (based on the MOVIELENS 1M dataset) and your recommendations must be based on cast members.
Follow this strict format and do not include any explanations, duplicates, or corrections in your response:
1. title (release year)
2. title (release year)
3. title (release year)
...
I've watched:
{movies}

Figura 1: *Prompt* para gerar recomendações. O campo `{movies}` é substituído pelos filmes assistidos pelo usuário.

Dado o caráter livre da linguagem natural gerada por LLMs, podem surgir desafios relacionados à padronização e interpretação das respostas. Para garantir a conformidade com o conjunto de dados a serem utilizados nos experimentos, nossa abordagem precisa de etapas de pré-processamento e filtragem. Um dos principais possíveis problemas é a diferença entre a descrição de itens na coleção e a descrição retornada pelo LLM. Por exemplo, em uma coleção de filmes, pode haver diferenças nas representações dos títulos dos filmes. Títulos iniciados com o artigo "The" são frequentemente registrados no conjunto de dados com o artigo ao final, como em "Godfather, The (1972)", enquanto o LLM normalmente retorna no formato canônico "The Godfather (1972)", o que pode gerar conflitos na validação. Assim, foi implementada uma etapa de normalização para tratar essas variações, assegurando que recomendações semanticamente equivalentes não fossem descartadas.

Em cenários onde os itens recomendados são baseados em descrições (e.g. filmes, pontos de interesse, etc.), outro tipo comum de erro está relacionado a pequenas variações na grafia dos itens, como "Pleasentville (1998)" e "Pleasantville (1998)" no cenário de filmes. Apesar de referirem-se ao mesmo filme, essas diferenças impedem o reconhecimento da correspondência. Para lidar com esses casos, propomos a utilização de um algoritmo de comparação de similaridade baseado na distância de *Levenshtein*, capaz de identificar e dimensionar discrepâncias ortográficas. Além desses casos, nossa abordagem também está preparada para lidar com outros tipos de erros, como a presença de itens com múltiplas descrições ou a repetição de recomendações para um mesmo usuário. Além disso, recomendações que não correspondessem a itens presentes no conjunto de dados foram descartadas. Por fim, o procedimento de solicitação é iterado até que sejam obtidas 10 recomendações válidas por usuário ou até o limite de cinco iterações consecutivas sem sucesso. Com o conjunto final de recomendações, realiza-se a comparação com o conjunto de teste individual de cada usuário, possibilitando a avaliação de desempenho do modelo. Desse modo, nossa abordagem permite validar as recomendações geradas pelo LLM da forma mais precisa possível, minimizando inconsistências decorrentes da linguagem natural.

3.1 Processo de Otimização de Prompt

A sensibilidade dos LLMs ao *prompt* é uma questão crucial que precisa ser considerada durante sua utilização, uma vez que *prompts* semanticamente semelhantes podem resultar em desempenhos significativamente diferentes [43]. Dessa forma, a engenharia de *prompt* tornou-se uma etapa fundamental para obter o melhor desempenho desses modelos. Para lidar com esse desafio, os autores de [43] propõem o método *OPRO* (***O**ptimization by **PRO**mpting*), que utiliza o próprio LLM para otimizar o *prompt* a ser utilizado em uma tarefa.

O *OPRO* transforma os LLMs em otimizadores iterativos baseados em linguagem natural. Nesse processo, o problema de otimização, como encontrar um *prompt* que maximize a acurácia em determinada tarefa, é descrito textualmente e apresentado ao modelo por meio de um *meta-prompt*. Esse *meta-prompt* inclui três elementos principais: uma descrição da tarefa, exemplos ilustrativos e um histórico das instruções previamente geradas com seus respectivos desempenhos. A cada iteração, o LLM propõe novas instruções, que são avaliadas e reincorporadas ao *meta-prompt*. O ciclo se repete até que não haja mais melhorias relevantes. Os resultados mostram que o *OPRO* gera *prompts* que superam instruções criadas manualmente.

Assim, inspirados nesse trabalho, propomos uma adaptação do método *OPRO* para o contexto de sistemas de recomendação. Nossa proposta parte de um *prompt* inicial, elaborado manualmente (e.g. Figura 1), cuja efetividade é avaliada por uma métrica específica. A partir dessa avaliação, iniciamos um processo iterativo de refinamento por meio de um *meta-prompt*, que inclui a descrição da tarefa, que consiste em gerar instruções para orientar a geração de recomendações, e um conjunto de instruções previamente criadas, acompanhadas de seus respectivos desempenhos. Esse processo também é conduzido pelo LLM *Llama3.1-8b-instruct*, quantizado em 4 *bits*. A cada iteração, o modelo recebe o *meta-prompt* como entrada e propõe novas instruções. Na primeira iteração, como ainda não há histórico, o *meta-prompt* inclui apenas a descrição da tarefa e o *prompt* inicial. A partir da segunda, as melhores instruções geradas anteriormente passam a ser incorporadas ao *meta-prompt*, enriquecendo o contexto fornecido ao modelo com o objetivo de que o refinamento ocorra progressivamente ao longo das etapas. A geração foi configurada para produzir até 1000 novos *tokens*. A amostragem foi ativada, com temperatura 1.6, *top-p* igual a 0.9 e *top-k* limitado a 40, controlando a aleatoriedade e a diversidade da saída.

O número total de iterações, a quantidade de novas instruções geradas em cada rodada e o número de instruções mantidas no *meta-prompt* são definidos como parâmetros do processo. Neste trabalho, realizamos seis iterações; em cada uma delas, o modelo gera quatro novas instruções, que são avaliadas individualmente e, em seguida, comparadas com as três melhores instruções já presentes no *meta-prompt*. A partir disso, as três com melhor desempenho são selecionadas para compor o *meta-prompt* da próxima iteração. A métrica utilizada na avaliação é a mesma aplicada ao *prompt* inicial e permanece constante ao longo de todo o processo.

Um aspecto fundamental da proposta é que as instruções geradas são utilizadas como conteúdo do campo `system` do *prompt* final. Assim, estamos otimizando o próprio contexto fornecido ao LLM, buscando instruções que o guiem da forma mais eficaz possível na geração das recomendações. Logo, a otimização ocorre sobre a formulação da tarefa e influencia diretamente o comportamento do modelo.

Para avaliar a eficácia do método, apresentamos os *meta-prompts* utilizados em duas estratégias de otimização distintas. Na primeira, adotamos a métrica *MAP* como critério principal, selecionando ao final das iterações a instrução que alcançou o maior valor nessa métrica. Na segunda estratégia, o *meta-prompt* é baseado na métrica *MAUT* [6], que agrega os indicadores *MAP*, *LTC*, *RMSE*, *NDCG@10* e *F1-score* em uma única medida composta (todas as métricas estão detalhadas na Seção 4.2). Para isso, todas as métricas foram normalizadas, com inversão do *RMSE* (uma vez que valores menores são preferíveis), e calculada sua média aritmética. Como o *MAUT* é sensível à composição do conjunto avaliado devido à normalização, ele foi recalculado a cada iteração considerando as sete instruções em avaliação (as três melhores previamente incluídas no *meta-prompt* e as quatro geradas na iteração atual) garantindo assim uma comparação justa e consistente. Ao final, selecionamos a instrução com maior valor de MAUT dentro da iteração em que essa métrica atingiu seu valor máximo, uma vez que, diferentemente do MAP, seus valores não são diretamente comparáveis entre iterações. A Figura 2 ilustra nossa proposta para o cenário de filmes.

System Prompt:
Your task is to create a clear and effective instruction for a movie recommendation task using the MovieLens 1M dataset. The instruction you generate will be used by a model to recommend movies based on user preferences, past ratings, or relevant contextual signals such as genre affinity, viewing history, or similar user behavior.

To help you write a stronger instruction, we provide examples of previous instructions along with quality scores. These are ordered from worst to best in terms of recommendation relevance, clarity, and alignment with the dataset. Use them as inspiration, but do not copy them.

{instructions}

Your instruction must explicitly include the following constraints:
- The output must be a numbered list in this exact format:
  1. title (release year)
  2. title (release year)
  3. title (release year)
  ...
- The model must output **only** the list — no additional text, explanations, or commentary.

To maximize recommendation quality:
- Encourage the model to consider patterns in user preferences, ratings, and genres.
- Make it clear that the recommendations should be personalized and relevant to the user context.
- Ensure that the instruction clearly connects the task to the MovieLens 1M dataset (which includes user ratings, genres, and timestamps).
- Encourage the model to suggest items that are both relevant and meaningfully ordered according to the user's preferences.
- Balance recommendations to not overly favor the most popular items, promoting a more diverse and personalized experience.
- Ensure that the instruction leads the model to capture deeper patterns in user behavior, including preferences across popularity levels and genre variety.
- The goal is to produce recommendations that are not only accurate, but also calibrated, diverse, and well-aligned with each user's unique profile.

Focus on crafting an instruction that is practical, specific, and optimized for generating relevant, high-quality recommendations in the correct format.
Focus on crafting an instruction that is practical, specific, and optimized for generating high-quality recommendations in the correct format.

User Prompt:
Create a new instruction that is concise and highly effective for the recommendation task using the MovieLens 1M dataset.
The instruction must explicitly state that the model's response should consist only of a numbered list of movie titles in the format:
1. title (release year)
2. title (release year)
3. title (release year)
...
It must also make clear that no additional text, explanations, or extra information should be included in the output.
Ensure that the instruction is presented directly, without mentioning that it is being generated or evaluated, and do not include any reference to performance metrics or scoring.

Figura 2: *Meta-prompt* para otimização, com trechos comuns e variações específicas destacadas para as métricas *MAP* (em azul) e *MAUT* (em laranja). O campo `{instructions}` é substituído, em cada iteração, pelas melhores instruções geradas até o momento, e suas respectivas pontuações.

Além das métricas, realizamos uma avaliação preliminar para observar o impacto do perfil dos usuários no processo de otimização. Em uma das versões, selecionamos aleatoriamente 100 usuários, sendo que 89 deles já recebiam 10 recomendações com o *prompt* inicial. Na outra, os 100 usuários escolhidos não atingiam esse mínimo. Essa diferenciação nos permitiu observar como tanto o critério de avaliação quanto as características do conjunto de usuários podem influenciar o resultado final. Ao todo, foram realizadas quatro execuções, variando entre a métrica utilizada e o perfil dos usuários selecionados, cada uma resultando em uma instrução final otimizada, utilizada posteriormente para a geração de recomendações.

4 AMBIENTE EXPERIMENTAL

Para responder às nossas perguntas de pesquisa, nesta seção detalhamos o ambiente experimental considerado em nossos experimentos.

4.1 Conjunto de Dados

O conjunto de dados utilizado neste trabalho é o MovieLens 1M [19], que originalmente contém 1.000.209 avaliações fornecidas por 6.040 usuários para aproximadamente 3.900 filmes distintos. Aplicamos sobre a coleção uma série de etapas de pré-processamento comumente usadas na literatura [14]. Primeiramente, removemos usuários com menos de 30 avaliações, garantindo dados suficientes para gerar recomendações significativas. Para cada usuário, selecionamos as 30 avaliações mais recentes, dividindo-as em conjuntos de treino e teste: as 10 avaliações mais recentes foram usadas para teste, e as 20 restantes para treino. Para garantir que todos os filmes do conjunto de teste também estivessem presentes no conjunto de treino, removemos quaisquer avaliações de teste para filmes que não apareciam nos dados de treino. Como resultado, o conjunto de dados final, após todo o pré-processamento, consiste em 158.446 avaliações — 105.780 no conjunto de treino e 52.666 no conjunto de teste — fornecidas por 5.289 usuários para 3.432 filmes distintos.

4.2 Métricas

Em nossos experimentos, avaliamos os efeitos de diferentes aspectos das recomendações em termos de precisão, cobertura, viés de popularidade e justiça, conforme detalhado abaixo:

4.2.1 Precisão. Utilizamos o *Normalized Discounted Cumulative Gain* (NDCG) para avaliar a qualidade da classificação das listas recomendadas. O NDCG compara o ganho cumulativo descontado do *ranking* produzido com o ganho de um *ranking* ideal, penalizando itens relevantes aparecendo em posições baixas [21]. O DCG no corte N é dado pela Eq. 1, normalizado pelo $IDCG@N$, que é o DCG obtido ao ordenar os itens por relevância decrescente (Eq. 2). O NDCG varia em $[0, 1]$, sendo 1 um *ranking* perfeito:

$$DCG@N = \sum_{i=1}^{N} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{1}$$

$$NDCG@N = \frac{DCG@N}{IDCG@N} \tag{2}$$

onde i é a posição do item na lista e rel_i é sua relevância na posição i.

A métrica *Mean Average Precision* (MAP) também é utilizada para avaliar a precisão das recomendações. O MAP varia de 0 a 1, onde valores mais altos são melhores e, conforme a Equação 3, mede a acurácia global, calculando a média da precisão ($AveP(i)$) para todos os itens recomendados.

$$MAP@N = \frac{1}{|N|} \sum_{i=1}^{N} AveP(i) \tag{3}$$

4.2.2 Justiça. Utilizamos uma métrica proposta por [13], chamada *Mean Rank Miscalibration* (MRMC), que abrange o intervalo $[0, 1]$, onde quanto menor, melhor. A Equação 4 obtém o valor de justiça, calculado a partir da divergência F entre as distribuições de gênero p e q no perfil de preferências e lista de recomendações, respectivamente. Como métrica de divergência, usamos a Kullback-Leibler [13], sendo que $F(p, q(\{\}))$ representa o pior caso de divergência da lista, usada para normalização. Na Equação 5, para cada usuário u é calculada a soma das médias dos valores dos erros de calibração, considerando cada posição da lista de recomendações. Por fim, a Equação 6 calcula o valor de RMRC para todos os usuários.

Em nossa proposta, consideramos os gêneros e a popularidade dos itens como atributos associados à miscalibração. Dessa forma, seguimos a estratégia de [14] ao usar a média harmônica (ou pontuação F1) entre MRMC de gêneros e popularidade[1], onde valores mais altos são melhores (Equação 7).

$$MC(p, q) = \frac{F(p, q)}{F(p, q(\{\}))} \tag{4}$$

$$RMC(u) = \frac{\sum_{i=1}^{N} MC(p, q(i))}{N} \tag{5}$$

$$MRMC = \frac{\sum_{u \in U} RMC(u)}{|U|} \tag{6}$$

$$F1 = 2\frac{(1 - MRMC\ Genre) * (1 - MRMC\ Pop)}{(1 - MRMC\ Genre) + (1 - MRMC\ Pop)} \tag{7}$$

4.2.3 Cobertura e Viés de Popularidade. Da mesma forma que [14], usamos a métrica *Long-Tail Coverage* (LTC) para medir a cobertura das recomendações e popularidade média do grupo *Group Average Popularity* (ΔGAP) [2] para medir o viés de popularidade das recomendações. A métrica LTC mede a proporção de itens únicos de cauda longa Φ efetivamente exposta nas recomendações[2]; varia de 0 (apenas itens populares) a 1 (apenas itens de nicho). Seja L_u a lista recomendada ao usuário u:

$$LTC = \frac{\left|(\bigcup_{u \in U} L_u) \cap \Phi\right|}{|\Phi|}. \tag{8}$$

Para o GAP, adotamos a mesma divisão em três grupos de usuários, com base em [1, 2] para o ΔGAP: **BlockBuster (BB)** cujo consumo dos usuários é de pelo menos 50% dos itens mais populares, **Nicho (N)** onde o consumo dos usuários é de pelo menos 50% dos itens de menor popularidade e **Diverso (D)** cujas preferências divergem dos outros dois grupos. A popularidade média dos itens recomendados a um grupo g é calculada pela Equação 9, onde ϕ representa a popularidade de um item e pu a lista de itens do perfil do usuário.

$$GAP(g) = \frac{\sum_{u \in g} \frac{\sum_{i \in pu} \phi(i)}{|pu|}}{|g|} \tag{9}$$

A partir disso, é possível representar o valor do GAP para o perfil dos usuários por meio de $GAP(g)_p$ e o GAP das recomendações por meio de $GAP(g)_r$. Assim, é possível calcular a variação de popularidade, como representada na Equação 10.

$$\Delta GAP(g) = \frac{GAP(g)_r - GAP(g)_p}{GAP(g)_p} \tag{10}$$

Valores negativos indicam redução do viés de popularidade, e vice-versa. Como os valores ótimos de ΔGAP devem ser próximos de zero, seguimos a estratégia de [35] ao utilizar o *Root Mean Squared Error* (RMSE) entre os três grupos de usuários, onde valores mais baixos são melhores:

$$RMSE = \frac{1}{3}\sqrt{\Delta GAP_{BB}^2 + \Delta GAP_N^2 + \Delta GAP_D^2} \tag{11}$$

4.2.4 Teoria da Utilidade Multiatributo (MAUT). A métrica MAUT permite condensar várias métricas de qualidade em um único escalar, facilitando a comparação global entre diferentes modelos de recomendação [6, 26].

Seja $M = \{MAP,\ NDCG,\ RMSE,\ LTC,\ F1\ Score\}$ o conjunto de métricas consideradas. Para cada modelo e para cada métrica $j \in M$, denotamos por m_{ij} o valor bruto obtido.

Como as métricas possuem escalas diferentes, primeiro aplicamos *min-max scaling* para torná-las comparáveis; se para alguma métrica os valores menores forem melhores, basta inverter o sinal antes de normalizar:

$$u_{ij} = \frac{m_{ij} - \min(m_j)}{\max(m_j) - \min(m_j)}, \tag{12}$$

onde $\min(m_j)$ e $\max(m_j)$ representam o pior e o melhor valor observado para a métrica j entre todos os métodos avaliados.

Definindo um vetor de pesos $\mathbf{w} = (w_1, \ldots, w_{|M|})$ tal que $\sum_{j=1}^{|M|} w_j = 1$, a utilidade global do método i é:

$$U_i = \sum_{j=1}^{|M|} w_j\, u_{ij}. \tag{13}$$

Quando não há preferência por nenhuma métrica específica, utiliza-se $w_j = \frac{1}{|M|}$, atribuindo importância igual a todas elas. Valores mais altos de U_i indicam melhor capacidade do método em equilibrar, de forma simultânea, todos os critérios de qualidade considerados. No nosso cenário, todas terão os mesmos pesos. A métrica U_i situa-se no intervalo $[0, 1]$. Assim, um método com $U_i \approx 1$ aproxima-se do melhor desempenho observado em todas as métricas, enquanto valores próximos de 0 revelam fraco desempenho global. Essa métrica permite uma avaliação abrangente e alinhada às questões de pesquisa, uma vez que reflete diretamente os *trade-offs* entre precisão, diversidade e justiça, aspectos centrais deste trabalho.

4.3 Métodos Tradicionais

Para avaliar os nossos modelos, selecionamos *baselines* que englobam diferentes abordagens de recomendação, do ranqueamento puramente popular aos métodos de calibração/diversidade e mitigação de viés de popularidade:

(1) **Popularity** [32]: Esse trabalho apresenta uma abordagem de calibração das recomendações com base na popularidade dos itens e no nível de interesse dos usuários por esse aspecto.
(2) **Personalized** [32]: Calibração personalizada das recomendações para cada usuário, o qual pode receber uma lista recomendada com base na proporção de interesse nos gêneros ou na popularidade dos itens, dependendo se o interesse dele na popularidade está acima de um limite definido.

[1]No caso da métrica MRMC aplicada à popularidade, substituímos cada gênero pelo grau de popularidade do item: nicho, diverso ou *blockbuster*, sendo tal classificação realizada com base no princípio de Pareto [2].
[2]Os itens de cauda longa são aqueles classificados como nicho ou diverso.

Modelo	MAP@10	NDCG@10	LTC	F1 Score	RMSE	MAUT
Métodos Tradicionais						
Popularity [32]	0,027	0,010	0,048	**0,539**	0,210	0,511
Personalized [32]	0,035	0,013	0,080	0,257	1,498	0,287
Steck [36]	0,037	0,013	0,075	0,243	1,130	0,320
Two-stage [35]	0,034	0,013	0,080	0,266	0,960	0,342
Abdoullahpouri [2]	0,026	0,010	0,046	0,537	1,115	0,412
BPR [29]	0,008	0,003	**0,542**	0,502	0,623	0,472
BPR Calibrado [14]	0,008	0,003	0,530	0,499	**0,044**	0,527
Modelo baseado em LLM (Llama)						
Sem otimização	**0,059**	**0,021**	0,517	0,296	0,367	**0,741**

Tabela 1: Comparação entre modelos tradicionais e o modelo baseado em LLM (LLaMa). Os melhores valores por métrica estão em negrito. Todos os resultados apresentaram significância estatística (p-*value* $< 0{,}05$), conforme o teste de Wilcoxon.

(3) **Steck** [36]: Trabalho clássico de calibração por conteúdo com base nos gêneros dos itens para alinhar a distribuição de categorias do ranking ao perfil do usuário.
(4) **Two-stage** [35]: Abordagem em duas etapas de calibração, aplicando primeiro a calibração por popularidade e depois a calibração com base nos gêneros, representando estratégias que combinam múltiplos objetivos de balanceamento.
(5) **Abdoullahpouri** [2]: *Baseline* inspirado em avaliações centradas no usuário para lidar com o viés de popularidade, incluído como representante de intervenções voltadas a reduzir a exposição desbalanceada entre itens populares e de nicho. Utiliza uma medida de divergência diferente da implementada pelo *baseline Popularity*.
(6) **BPR** [29]: Método tradicional de *pairwise ranking* para *feedback* implícito; usado como referência consolidada de recomendações personalizadas.
(7) **BPR Calibrado** [14]: Alteração do BPR com etapa adicional de processamento para mitigar o viés de popularidade; incluída para avaliar o efeito da calibração sobre um modelo que aprimora o BPR e calibra as recomendações com base na popularidade dos itens.

5 RESULTADOS

Nesta seção, apresentamos os resultados obtidos, divididos em duas subseções, cada uma diretamente relacionada às questões de pesquisa previamente definidas. Todos os experimentos relatados nesta seção foram executados com seis repetições visando garantir a consistência nos resultados obtidos. Para assegurar a robustez das comparações, aplicamos o teste estatístico não paramétrico de Wilcoxon [30], com nível de significância de 5%, para verificar se as diferenças entre os métodos são estatisticamente significativas.

5.1 RQ1: LLM versus Métodos Tradicionais

Com relação às métricas de **precisão** (MAP e NDCG@10), a estratégia baseada em LLM apresentou ganhos expressivos em comparação aos métodos tradicionais, conforme ilustrado na Tabela 1. O MAP médio dos métodos tradicionais variou entre 0,008 e 0,037, ao passo que a estratégia LLM atingiu 0,059. Resultados similares foram observados para o NDCG@10, com valores entre 0,003 e 0,013 nos métodos tradicionais e 0,021 na estratégia LLM. Esses números indicam que a estratégia baseada em LLM é mais eficaz em posicionar itens relevantes nas primeiras posições da lista de recomendação, fator crítico para a experiência do usuário.

No que tange à **diversidade** e **cobertura**, mensuradas pela métrica LTC, a estratégia LLM também se destacou. Embora alguns métodos tradicionais, como o BPR e BPR Calibrado, tenham alcançado os melhores valores de LTC, o LLM ainda obteve um desempenho elevado (0,517), demonstrando sua capacidade de conciliar precisão com maior variedade e alcance nas recomendações, equilibrando adequadamente o *trade-off* entre precisão e diversidade [44]. Isso evidencia seu potencial para mitigar a concentração em itens populares, ampliando a exposição a conteúdos diversos.

Quanto à métrica **RMSE**, que avalia o desequilíbrio de popularidade entre grupos (quanto menor, melhor), o LLM superou a maioria dos métodos tradicionais, cujos valores oscilaram entre 0,044 e 1,498. Embora o BPR Calibrado e o Popularity tenham registrado RMSE inferiores, ambos apresentaram sérios compromissos em outras dimensões: o BPR Calibrado teve os menores valores de MAP e NDCG@10, enquanto o Popularity exibiu cobertura limitada e precisão modesta (MAP de 0,027). Já o LLM conseguiu manter um RMSE competitivo sem abrir mão da qualidade do ranking e da diversidade, como refletido também em seu alto valor de MAUT.

No aspecto da **justiça**, o LLM apresentou desempenho inferior aos modelos tradicionais, com um F1 Score de 0,296, ante valores entre 0,243 e 0,539 nos demais métodos. Esse resultado ilustra o conhecido *trade-off* entre diversidade e justiça [38, 46]. Ainda assim, ao se considerar a métrica MAUT — que integra múltiplas dimensões como precisão, diversidade e justiça — o modelo LLM demonstrou desempenho notavelmente superior. Enquanto os métodos tradicionais variaram entre 0,287 e 0,527, o LLM alcançou 0,741, sinalizando uma solução mais equilibrada e robusta.

Assim, em resposta à RQ1, os resultados indicam que estratégias baseadas em LLM superam consistentemente os métodos tradicionais em precisão e diversidade, mantendo níveis competitivos de justiça. Sua capacidade de otimizar múltiplos objetivos simultaneamente torna-as alternativas mais eficazes e balanceadas para sistemas de recomendação modernos.

5.2 RQ2: Efeitos da Otimização de *Prompts*

5.2.1 ***Prompts Resultantes da Otimização.*** Conforme descrito na Seção 3.1, aplicamos o processo de otimização em diferentes

configurações, gerando *prompts* otimizados para cada cenário. As variações consideraram o uso das métricas *MAP* e *MAUT*, bem como diferentes conjuntos de usuários: *MAP* com 100 usuários aleatórios (*MAP_random* - Figura 3); *MAP* com 100 usuários com menos de 10 recomendações (*MAP_below_10* - Figura 4); *MAUT* com 100 usuários aleatórios (*MAUT_random* - Figura 6); e *MAUT* 100 usuários com menos de 10 recomendações (*MAUT_below_10* - Figura 5). Em todos os casos, o conteúdo do campo `User Prompt` utilizado para solicitar as recomendações ao modelo foi o mesmo apresentado na Figura 1, garantindo assim consistência na forma de interação do usuário com o modelo ao longo dos diferentes experimentos.

System Prompt:
Generate Personalized Movie Recommendations Using the MovieLens 1M Dataset

Use the MovieLens 1M dataset to generate a list of personalized movie recommendations based on the user's context (if provided). This list should consider patterns in user ratings, genres, and timestamps to create unique and relevant suggestions tailored to each viewer.

Your response should be a numbered list in the exact format:

1. Title (release year)
2. Title (release year)
3. Title (release year)
...

This list **must consist only of**:

* Numbered movie titles with release years in the specified format
* Exactly one title per entry
* Release years included for each title

Do not include:

* Any additional text, explanations, or comments in your response
* Extra information or unnecessary details about the movies or the users
* Non-movie titles, incomplete, or missing metadata

Provide your output exactly as specified above, following the specified format for each entry.

Figura 3: *Prompt* gerado pelo processo de otimização com a métrica *MAP* utilizando 100 usuários aleatórios.

System Prompt:
Recommend Movies from the MovieLens 1M Dataset.

Given the MovieLens 1M dataset and a relevant user context, output a **complete** list of personally recommended movies in the following format:

1. title (release year)
2. title (release year)
3. title (release year)
...

Your response should consist **solely** of this numbered list. Please adhere strictly to the specified format.

You may draw upon patterns in user ratings, genres, preferences, and contextual clues from the dataset to generate your recommendations.

Output the recommendation list exactly as instructed; ensure that it includes the title and release year for each movie.

Figura 4: *Prompt* gerado pelo processo de otimização com a métrica *MAP* utilizando 100 usuários com menos de 10 recomendações.

5.2.2 ***Impactos da Otimização de Prompts no Desempenho do LLM.*** Após demonstrarmos, em RQ1, que a estratégia baseada em LLM (LLaMa) supera as abordagens tradicionais em termos de balanceamento entre as métricas de precisão, diversidade e justiça, avançamos para investigar se e como a otimização de *prompts* pode melhorar ainda mais o desempenho desses modelos.

Iniciamos nossa análise nas métricas de precisão (MAP e NDCG@10). Tomando o LLM sem otimização como referência (MAP = 0,059; NDCG@10 = 0,021), na Tabela 2, as variantes *MAP_below_10* e *MAUT_below_10* apresentaram queda estatisticamente significativa em MAP (0,056 e 0,055), enquanto *MAP_random* e *MAUT_random*

System Prompt:
Recommend Movie Titles Based on User Behavior

Respond with a numbered list of recommended movie titles and release years.

The format for the response is:

1. title (release year)
2. title (release year)
3. title (release year)
...

Include the exact movie title and release year for each item.

The response must be a standalone numbered list of movie titles and dates. Provide only this list - do not include any additional text or information.

Figura 5: *Prompt* gerado pelo processo de otimização com a métrica *MAUT* utilizando 100 usuários com menos de 10 recomendações.

System Prompt:
Personalized Movie Recommendation

Your task is to generate a list of movie recommendations based on user preferences and past ratings from the MovieLens 1M dataset. Each movie should be accompanied by its release year. Follow these rules exactly:

* Your response should consist only of a numbered list of movie titles, one movie per line, in the exact format:
1. title (release year)
2. title (release year)
3. title (release year)
...

* The list should contain multiple movie entries; there is no minimum or maximum length for the list.
* Leave no blank lines between the list entries.
* Do not include any additional text, headings, or information in your response; the list should be the only thing present.
* Each title should be a string enclosed in parentheses, indicating its release year, or left uncaptioned but present only as part of a title. For example:
1. "Inception (2010)"
2. "The Dark Knight (2008)"

* Provide movie titles that accurately reflect the preferences and habits of the users in the MovieLens 1M dataset.

Ensure your recommendations are calibrated and balance diverse film genres, as well as offer unique profiles that account for relevance, alignment with user profiles, and genre affinity.

Include only these necessary items:

1. title (release year)
2. title (release year)
3. title (release year)
...

Your response will take the form of a numbered list with the movie titles presented in the above format, offering a set of personalized and well-calibrated movie suggestions for viewers interested in a varied selection of films from the MovieLens dataset.

Figura 6: *Prompt* gerado pelo processo de otimização com a métrica *MAUT* utilizando 100 usuários aleatórios.

foram estatisticamente equivalentes. Para o NDCG@10, houve um empate estatístico para as diferentes otimizações, com exceção de *MAP_below_10* (0,019), que apresentou redução estatisticamente significativa. Em síntese, é possível otimizar sem degradar a precisão quando se utiliza *MAP_random* ou *MAUT_random*.

Em relação ao viés de popularidade, utilizando a métrica RMSE (menor é melhor), o LLM sem otimização obteve 0,367. As estratégias *MAP_random* (0,490) e *MAUT_random* (0,433) apresentaram valores estatisticamente inferiores, ao passo que *MAP_below_10* (0,350) e *MAUT_below_10* (0,406) apresentaram empate estatístico com LLM sem otimização. Em outras palavras, otimizações "*random*" tendem a aumentar o erro agregado entre grupos, enquanto os processos "*below_10*" preservam esse equilíbrio.

Modelo	MAP@10	NDCG@10	LTC	F1 Score	RMSE	MAUT
Modelos LLM (LLaMA) com otimização						
MAP_below_10	0,056 ▼	0,019 ▼	**0,592** ▲	0,292 ●	**0,350** ●	0,392 ▼
MAP_random	0,057 ●	**0,021** ●	0,512 ●	0,300 ●	0,490 ▼	0,458 ●
MAUT_below_10	0,055 ▼	0,020 ●	0,583 ▲	0,297 ●	0,406 ●	0,445 ●
MAUT_random	0,055 ●	0,020 ●	0,486 ▼	**0,302** ●	0,433 ▼	0,401 ●
Modelo LLM (LLaMA) sem otimização						
Sem otimização	**0,059**	**0,021**	0,517	0,296	0,367	**0,518**

Tabela 2: Comparação entre o modelo LLM (LLaMa) sem otimização e quatro variações com diferentes estratégias de otimização. Os melhores valores em cada métrica estão em negrito. O símbolo ▲ indica que a otimização apresentou melhora estatisticamente significativa em relação ao modelo sem otimização (p-*value* < 0.05, teste de Wilcoxon); ● indica ausência de diferença significativa; e ▼ indica que o modelo sem otimização foi estatisticamente superior.

Os processos *MAP_below_10* e *MAUT_below_10* elevaram o LTC de forma significativa (0,592 e 0,583) em relação ao *baseline* (0,517), reforçando que esses processos de otimização aumentam a diversidade das recomendações. Já as otimizações *MAP_random* (0,512) e *MAUT_random* (0,486) apresentaram empate e perdas estatísticas, respectivamente. Não houve diferenças estatisticamente significativas entre a estratégia baseada em LLM e suas variantes otimizadas em termos de justiça, o que mostra que as otimizações não afetaram a justiça do sistema em termos de gênero e popularidade.

Em relação à métrica MAUT, as estratégias *MAP_random* (0,458), *MAUT_random* (0,401) e *MAUT_below_10* (0,445) não apresentaram alterações significativas para o modelo não otimizado, enquanto *MAP_below_10* (0,392) teve redução estatisticamente significativa.

Em essência, há espaço para otimizar sem afetar o sistema como um todo e ainda há a possibilidade de ganhos em termos de diversidade e justiça. Os resultados mostram que os processos podem ser selecionados conforme o objetivo do sistema. Se o objetivo é aumentar diversidade/cobertura e reduzir miscalibração, *MAP_below_10* e *MAUT_below_10* são preferíveis, aceitando-se um *trade-off* de precisão/MAUT. Se a prioridade é preservar precisão e qualidade global com ganhos em calibração, *MAP_random* é a melhor opção. Por fim, *MAUT_random* é uma opção quando se busca estabilidade geral, mas não é indicada quando LTC e RMSE são preferências centrais.

Em suma, respondendo à RQ2, os resultados demonstram que a otimização de *prompts* permite ajustar o sistema de recomendação a diferentes objetivos, embora não produza ganhos uniformes em todas as métricas. Algumas estratégias mantêm altos níveis de precisão, mas com prejuízos em termos de equidade, enquanto outras favorecem a diversidade de exposição, ainda que com uma leve redução no desempenho. Portanto, a seleção da estratégia de otimização deve ser guiada pelas prioridades específicas de cada contexto de aplicação, considerando os *trade-offs* envolvidos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, investigamos o desempenho de sistemas de recomendação, baseados em LLMs, em comparação com métodos tradicionais, bem como os efeitos de diferentes estratégias de otimização de *prompts*. A avaliação foi conduzida com rigor estatístico, utilizando múltiplas métricas que contemplam precisão, diversidade, justiça, equilíbrio entre grupos e desempenho agregado.

Os resultados indicam que estratégias baseadas em LLM, mesmo sem otimizações específicas, apresentam desempenho superior à maioria dos métodos tradicionais, especialmente em termos de precisão e equilíbrio geral. Além disso, demonstra boa capacidade de diversificação das recomendações e cobertura de itens, mantendo um desempenho competitivo também no controle de discrepâncias entre grupos. Embora alguns métodos tradicionais tenham obtido valores melhores em métricas pontuais, como o RMSE, esses ganhos geralmente vieram acompanhados de perdas significativas em outras dimensões, como precisão ou diversidade. Dessa forma, estratégias baseadas em LLM se destacam por oferecer um equilíbrio mais favorável entre os diferentes objetivos do sistema. Por outro lado, as estratégias de otimização de *prompts* não proporcionaram melhorias universais, mas funcionam como mecanismos eficazes para calibrar as prioridades do sistema. Estratégias que buscam preservar a precisão mostraram desempenho sólido nas métricas clássicas de ranqueamento, embora com aumento nas discrepâncias entre grupos. Por outro lado, abordagens voltadas à ampliação da diversidade conseguiram promover maior exposição à cauda longa, ainda que com leves perdas em precisão. A métrica de justiça apresentou variações discretas entre as estratégias, e o desempenho agregado foi inferior principalmente nas estratégias que não consideraram limites inferiores nas recomendações. Esses achados reforçam que a escolha da estratégia de otimização deve estar alinhada aos objetivos específicos da aplicação, em vez de se pautar por expectativas de ganhos uniformes.

Como direções para pesquisas futuras, destacam-se três frentes principais. Primeiramente, propomos a realização de experimentos em ambientes *online*, por meio de testes A/B, para avaliar o impacto das recomendações em métricas de engajamento e satisfação dos usuários. Em segundo lugar, expandiremos a análise para outros domínios e conjuntos de dados, incluindo contextos com diferentes graus de popularidade e esparsidade, a fim de verificar a robustez dos achados. Por fim, exploraremos variações arquiteturais de LLMs e novas estratégias de construção e otimização de *prompts*, assim como métricas mais refinadas de justiça e impacto social, visando aprofundar a compreensão sobre os limites e potencialidades desses modelos em sistemas de recomendação multiobjetivo.

AGRADECIMENTOS

Financiado por CNPq, INCT-TILD-IAR, Fapesp, Fapemig e AWS.

REFERÊNCIAS

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The Unfairness of Popularity Bias in Recommendation. *CoRR* abs/1907.13286 (2019). arXiv:1907.13286 http://arxiv.org/abs/1907.13286

[2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward C. Malthouse. 2021. User-centered Evaluation of Popularity Bias in Recommender Systems. In *Proceedings of the 29th ACM Conference UMAP 2021, Utrecht*. ACM, 119–129. doi:10.1145/3450613.3456821

[3] Paul Dany Flores Atauchi, André Levi Zanon, Leonardo Chaves Dutra da Rocha, and Marcelo Garcia Manzato. 2025. Do Calibrated Recommendations Affect Explanations? A Study on Post-Hoc Adjustments. 16 (Jun. 2025), 441–460. doi:10.5753/jis.2025.5563

[4] James Bennett and Stan Lanning. 2007. The netflix prize. (2007).

[5] Guilherme Bittencourt, Guilherme Fonseca, Yan Andrade, Nícollas Silva, and Leonardo Rocha. 2023. A survey on review-aware recommendation systems. In *Proceedings of the 29th Brazilian Symposium on Multimedia and the Web*. 198–207.

[6] Rodrigo Carvalho and Leonardo Rocha. 2020. Estratégias para Aprimorar a Diversidade Categórica e Geográfica de Sistemas de Recomendação de POIs. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*. SBC, 23–26.

[7] Oscar Celma and Paul Lamere. 2011. Music recommendation and discovery revisited. In *Proceedings of the fifth ACM RecSys*. 7–8.

[8] Luiz Chaves, Nícollas Silva, Rodrigo Carvalho, Adriano C. M. Pereira, and Leonardo Rocha. 2019. Exploiting the user activity-level to improve the models' accuracy in point-of-interest recommender systems. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web (WebMedia '19)*. 341–348. doi:10.1145/3323503.3349551

[9] Chien Chin Chen, Shun-Yuan Shih, and Meng Lee. 2016. Who should you follow? Combining learning to rank with social influence for informative friend recommendation. *Decision Support Systems* 90 (2016), 33–45.

[10] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. 2003. Is seeing believing? How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI*. 585–592.

[11] Washington Cunha, Leonardo Rocha, and Marcos André Gonçalves. 2025. A thorough benchmark of automatic text classification: From traditional approaches to large language models. *arXiv preprint arXiv:2504.01930* (2025).

[12] Diego Corrêa da Silva and Dietmar Jannach. 2025. Calibrated Recommendations: Survey and Future Directions. https://arxiv.org/abs/2507.02643

[13] Diego Corrêa da Silva, Marcelo Garcia Manzato, and Frederico Araújo Durão. 2021. Exploiting personalized calibration and metrics for fairness recommendation. *Expert Systems with Applications* 181 (2021), 115112. doi:10.1016/j.eswa.2021.115112

[14] Rodrigo Ferrari de Souza and Marcelo Garcia Manzato. 2024. Uma Abordagem em Etapa de Processamento para Redução do Viés de Popularidade. In *Brazilian Symposium on Multimedia and the Web (WebMedia)*. SBC, 310–317.

[15] Guilherme Fonseca, Washington Cunha, Gabriel Prenassi, Marcos André Gonçalves, and Leonardo Chaves Dutra Da Rocha. 2025. Instance-Selection-Inspired Undersampling Strategies for Bias Reduction in Small and Large Language Models for Binary Text Classification. In *Proceedings of the 63rd ACL*. Association for Computational Linguistics, 9323–9340.

[16] Guilherme Fonseca, Gabriel Prenassi, Washington Cunha, Marcos André Gonçalves, and Leonardo Rocha. 2024. Estratégias de Undersampling para Redução de Viés em Classificação de Texto Baseada em Transformers. In *Brazilian Symposium on Multimedia and the Web (WebMedia)*. SBC, 144–152.

[17] Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025. LLM4Rerank: LLM-based Auto-Reranking Framework for Recommendations. In *Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. 228–239. doi:10.1145/3696410.3714922

[18] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015).

[19] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. doi:10.1145/2827872

[20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[21] Aryan Jadon and Avinash Patil. 2024. A comprehensive survey of evaluation techniques for recommendation systems. In *International Conference on Computation of Artificial Intelligence & Machine Learning*. Springer, 281–304.

[22] Dietmar Jannach. 2022. Multi-Objective Recommender Systems: Survey and Challenges. arXiv:2210.10309 [cs.IR] https://arxiv.org/abs/2210.10309

[23] Anastasiia Klimashevskaia, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. 2024. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction* 34, 5 (2024), 1777–1834.

[24] Jan Malte Lichtenberg, Alexander Buchholz, and Pola Schwöbel. 2024. Large Language Models as Recommender Systems: A Study of Popularity Bias. In *Proceedings of the SIGIR 2024 Workshop on Generative Information Retrieval*.

[25] Dairui Liu, Boming Yang, Honghui Du, Derek Greene, Aonghus Lawlor, Ruihai Dong, and Irene Li. 2023. Recprompt: A prompt tuning framework for news recommendation using large language models. *CoRR* (2023).

[26] Silvia Beatriz Neiva and Luiz Flavio Autran Monteiro Gomes. 2007. A aplicação da teoria da utilidade multiatributo à escolha de um software de e-procurement. *Revista Tecnologia* 28, 2 (2007).

[27] Gustavo Mendonça Ortega, Rodrigo Ferrari de Souza, and Marcelo Garcia Manzato. 2024. Evaluating Zero-Shot Large Language Models Recommenders on Popularity Bias and Unfairness: A Comparative Approach to Traditional Algorithms. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*. SBC, 45–48.

[28] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM computing surveys (CSUR)* 51, 4 (2018), 1–36.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[30] Denise Rey and Markus Neuhäuser. 2011. Wilcoxon-signed-rank test. In *International encyclopedia of statistical science*. Springer, 1658–1659.

[31] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. *Recommender systems handbook* (2015), 1–34.

[32] Andre Sacilotti, Rodrigo Ferrari de Souza, and Marcelo Garcia Manzato. 2023. Counteracting popularity-bias and improving diversity through calibrated recommendations. In *In Proceedings of the 25th International Conference on Enterprise Information Systems*, Vol. 1. Scitepress, Prague, Czech Republic.

[33] Nícollas Silva, Heitor Werneck, Thiago Silva, Adriano CM Pereira, and Leonardo Rocha. 2021. A contextual approach to improve the user's experience in interactive recommendation systems. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*. 89–96.

[34] Rodrigo Souza and Marcelo Manzato. 2024. Explorando Formas de Calibração e Redução do Viés de Popularidade em Sistemas de Recomendação. In *Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web* (Juiz de Fora/MG). SBC, Porto Alegre, RS, Brasil, 9–10. doi:10.5753/webmedia_estendido.2024.244380

[35] Rodrigo Souza and Marcelo Manzato. 2024. A Two-Stage Calibration Approach for Mitigating Bias and Fairness in Recommender Systems. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. ACM, New York, NY, USA.

[36] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*. 154–162.

[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[38] Célina Treuillier, Sylvain Castagnos, Özlem Özgöbek, and Armelle Brun. 2024. Beyond Trade-offs: Unveiling Fairness-Constrained Diversity in News Recommender Systems *(UMAP '24)*. 143–148. doi:10.1145/3627043.3659571

[39] Jiawei Wang, Xinyu Chen, Kuan-Chieh Lee, Deb Ghosh, Neelesh Rao, and Hexiang Hu. 2025. Automating Personalization: Prompt Optimization for Recommendation Reranking. arXiv:2504.03965 [cs.IR]

[40] Heitor Werneck, Nícollas Silva, Matheus Carvalho Viana, Fernando Mourão, Adriano C. M. Pereira, and Leonardo Rocha. 2020. A Survey on Point-of-Interest Recommendation in Location-based Social Networks. In *Proceedings of the Brazilian Symposium on Multimedia and the Web* (São Luís, Brazil) *(WebMedia '20)*. Association for Computing Machinery, New York, NY, USA, 185–192. doi:10.1145/3428658.3430970

[41] Heitor Werneck, Nícollas Silva, Matheus Carvalho Viana, Fernando Mourão, Adriano CM Pereira, and Leonardo Rocha. 2020. A survey on point-of-interest recommendation in location-based social networks. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*. 185–192.

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

[43] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

[44] André L. Zanon, Leonardo Chaves Dutra da Rocha, and Marcelo Garcia Manzato. 2022. Balancing the trade-off between accuracy and diversity in recommender systems with personalized explanations based on Linked Open Data. *Knowl. Based Syst.* 252 (2022), 109333. doi:10.1016/J.KNOSYS.2022.109333

[45] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*.

[46] Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu C. Aggarwal, and Tyler Derr. 2025. Fairness and Diversity in Recommender Systems: A Survey. *ACM Trans. Intell. Syst. Technol.* 16, 1, Article 2 (Jan. 2025), 28 pages. doi:10.1145/3664928

[47] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*. PMLR, 12697–12706.