Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

Leveraging LLMs for Explainable Recommender Systems: Exploring User Perceptions and Faithfulness in Generated Explanations

Ítallo de Sousa Silva

Campina Grande, Paraíba, Brasil
Fevereiro/2025

Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Coordenação de Pós-Graduação em Ciência da Computação

Leveraging LLMs for Explainable Recommender Systems: Exploring User Perceptions and Faithfulness in Generated Explanations

Ítallo de Sousa Silva

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Metodologia e Técnicas da Computação

Leandro Balby Marinho

(Orientador)

Campina Grande, Paraíba, Brasil

S586l Silva, Ítallo de Sousa.

Leveraging LLMs for explainable recommender systems : exploring user perceptions and faithfulness in generated explanations / Ítallo de Sousa Silva. – Campina Grande, 2025.

85 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2025.

"Orientação: Prof. Dr. Leandro Balby Marinho".

Referências.

1. Grandes Modelos de Linguagem (LLMs). 2. Sistemas de Recomendação (RSs). 3. Estudo de Usuário. 4. Recuperação de Informação. I. Marinho, Leandro Balby. II. Título.

CDU 004.434:004.8(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA MEIRE EMANUELA DA SILVA MELO CRB-15/568

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secpg@computacao.ufcg.edu.br

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

ÍTALLO DE SOUSA SILVA

LEVERAGING LLMS FOR EXPLAINABLE RECOMMENDER SYSTEMS: EXPLORING USER PERCEPTIONS AND FAITHFULNESS IN GENERATED EXPLANATIONS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 13/02/2025

Prof. Dr. LEANDRO BALBY MARINHO, Orientador, UFCG

Prof. Dr. CLÁUDIO ELÍZIO CALAZANS CAMPELO, Examinador Interno, UFCG

Prof. Dr. ADRIANO ALONSO VELOSO, Examinador Externo, UFMG

Documento assinado eletronicamente por **LEANDRO BALBY MARINHO**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 14/02/2025, às 16:44, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **CLAUDIO ELIZIO CALAZANS CAMPELO**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 17/02/2025, às 12:15, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **5231074** e o código CRC **13B55E5D**.

Resumo

Sistemas de recomendação (RSs) tornaram-se comuns no dia a dia de boa parte da população, auxiliando usuários na descoberta de itens relevantes em diversos domínios. No entanto, a crescente complexidade dos RSs levanta preocupações sobre sua transparência e interpretabilidade, especialmente em aplicações de alto impacto. Esta dissertação investiga o potencial dos Grandes Modelos de Linguagem (LLMs) para gerar explicações automatizadas e centradas no ser humano para RSs e avalia sua fidelidade em refletir o raciocínio interno dos modelos. Avaliamos recomendações personalizadas de filmes e explicações geradas pelo GPT-3.5 Turbo por meio de um estudo com usuários, medindo eficácia, personalização e poder de persuasão. Um estudo complementar, abrangendo recomendações de filmes, músicas e livros geradas por quatro LLMs (a saber, GPT-4o, Llama3, Gemma2 e Mixtral 8x7B), avaliou a fidelidade dessas explicações usando uma avaliação axiomática baseada no Acordo de Importância de Características. Nossos resultados revelaram que, embora as recomendações geradas pelos LLMs tenham melhorado a satisfação do usuário em comparação com seleções aleatórias, as explicações frequentemente não atendiam aos critérios de fidelidade. Surpreendentemente, explicações baseadas em preferências do usuário não foram consistentemente percebidas como mais personalizadas, eficazes ou persuasivas do que explicações genéricas. As principais contribuições incluíram uma avaliação centrada no usuário da qualidade das explicações, um método axiomático para avaliar a fidelidade, percepções sobre preferências dos usuários e tipos de explicações, além de uma análise da interação entre os objetivos das explicações. Desafios notáveis identificados incluem as capacidades limitadas de personalização dos LLMs, a variabilidade nos resultados devido ao comportamento não determinístico e a natureza inerentemente de caixa-preta desses modelos. Este trabalho destaca as promessas e limitações dos LLMs em RSs Explicáveis e fornece uma base para futuras pesquisas que busquem melhorar o alinhamento entre a percepção do usuário e a fidelidade das explicações.

Abstract

Recommender systems (RSs) have become ubiquitous, assisting users in discovering relevant items across various domains. However, the increasing complexity of RSs raises concerns about their transparency and interpretability, particularly in high-stakes applications. This thesis investigates the potential of Large Language Models (LLMs) to generate automated, human-centered explanations for RSs and assesses their faithfulness in reflecting the models' internal reasoning. We evaluated personalized movie recommendations and explanations generated by GPT-3.5 Turbo through a user study, measuring effectiveness, personalization, and persuasiveness. A follow-up study across movie, song, and book recommendations generated by four LLMs (namely, GPT-4o, Llama3, Gemma2, and Mixtral 8x7B) assessed the faithfulness of these explanations using an axiomatic evaluation based on the Feature Importance Agreement. Our findings revealed that while LLM-generated recommendations improved user satisfaction compared to random selections, the explanations often failed to meet faithfulness criteria. Surprisingly, explanations based on user preferences were not consistently perceived as more personalized, effective, or persuasive than generic explanations. Key contributions included a user-centric evaluation of explanation quality, an axiomatic method for assessing faithfulness, insights into user preferences and explanation types, and an analysis of the interplay between explanation goals. Notable challenges identified include LLMs' limited personalization capabilities, variability in outputs due to non-deterministic behavior, and the inherent black-box nature of these models. This work highlights the promise and limitations of LLMs in Explainable RSs and provides a foundation for future research to enhance the alignment between user perception and explanation faithfulness.

Agradecimentos

A conclusão desta dissertação foi um esforço coletivo e não teria sido possível sem o apoio e a colaboração de muitas pessoas. Primeiramente, agradeço aos meus pais, Zélia e Vanderlei, pelo constante apoio em todas as etapas da minha vida. Sua dedicação e confiança foram essenciais para que eu pudesse alcançar mais este objetivo.

Agradeço ao meu orientador, Prof. Leandro Marinho, pela orientação, incentivo e paciência durante todo o desenvolvimento desta dissertação. Estendo também minha gratidão aos professores Alain Said (University of Gothenburg) e Martijn Willemsen (TU Eindhoven & JADS) pela colaboração no artigo *Leveraging ChatGPT for Automated Human-centered Explanations in Recommender Systems*, fundamental para o desenvolvimento deste trabalho, e ao Prof. Andrey Brito pela ajuda no custeio da viagem para a apresentação do respectivo artigo.

Agradeço ainda a meu namorado, Cézar, pelo carinho, compreensão e apoio ao longo desta caminhada. Sua presença e incentivo constante foram fundamentais nos momentos desafiadores e tornaram essa jornada muito mais leve e significativa. Sou imensamente grato por ter você ao meu lado. E a meus amigos, agradeço pelas palavras de incentivo e pelos momentos de descontração que me ajudaram a manter o equilíbrio ao longo desta jornada.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e da Bolsa 09/2021 da Fundação de Apoio à Pesquisa do Estado da Paraíba (FAPESQ).

Contents

Glossary

CoT chain-of-though reasoning

ICL In-Context Learning

LLM Largue Language Model

LLM-Rec LLM-based Recommender System

NLP Natural Language Processing

RQ research question

RS Recommender System

SHAP SHapley Additive exPlanations

XAI Explainable Artificial Intelligence

List of Figures

List of Tables

Chapter 1

Introduction

Recommender Systems (RSs) have become an integral part of our daily lives. Their presence can be observed in various domains, such as e-commerce and media streaming, where they aim to help users find relevant items, i.e., items aligned with their preferences. However, to obtain more accurate results, these models are becoming increasingly complex, raising concerns about their transparency and interpretability (or explainability).

A transparent and interpretable RS should provide information about how and why a recommendation was made for a particular user. These properties are fundamental when applying these models to high-risk scenarios, such as finance, law, and healthcare, where wrong decisions can substantially harm people's or enterprises' rights and interests. Significant efforts have been made to develop explainable RSs aligned with these properties [53]. Advances in this area would help users make well-informed decisions and organizations comply with data protection and ethical regulations, such as the LGPD in Brazil[1] and the European Union Artificial Intelligence Act[2].

Among the possibilities for generating explainable RSs, one approach is to use human-centered explanations, i.e., explanations in natural language that approximate how people explain things to each other. Several works that explored this topic observed a positive impact on user perception when the explanation is more human-like. However, for many years, this kind of approach has relied mainly on humans to generate the recommendations and explanations, making this type of explanation cost-ineffective in terms of time and money

[1] `https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709compilado.htm`

[2] `https://artificialintelligenceact.eu/`

and not scalable for large bases of users and items [3, 7, 22]. Thus, there is a strong need for novel scalable techniques to generate human-centered explanations, such as Largue Language Models (LLMs).

LLMs have empowered services like ChatGPT to provide human-like conversational experiences up to the point where an untrained individual cannot distinguish between model and human [10]. These models have also demonstrated impressive capabilities by successfully performing tasks such as sentiment analysis, question answering, and even recommendations, using only a few examples (few-shot) or even none at all (zero-shot) as input, without requiring any additional training [5].

Research on RSs powered by LLMs has demonstrated that LLMs like ChatGPT can make recommendations and provide explanations using only a few historical data from the user [21, 24, 49, 55]. Including some information about user historical preferences in a prompt to these models allows them to generate personalized recommendations accompanied by personalized explanations, which is possible due to the knowledge acquired by these models from training on large-scale web corpora.

Figure 1.1 depicts two examples of interactions with ChatGPT (with GPT-4o[3]), in the first (left) the user asks for a movie recommendation without giving much information. In contrast, in the second (right), the user includes information about their favorite movies. We can see that in both cases, the model was able to provide a recommendation along with a human-like explanation, which in the first case highlighted details about the plot and the widespread impression of the movie and in the second included a comparison with each favorite movie in the user's input. Despite these remarkable capabilities, at the start of this research, few works had dived into understanding the users' perception of these LLM-generated explanations, i.e., how users experience generic versus personalized recommendations and explanations. In particular, no work had been done for the recommendation task.

Therefore, our research evaluated users' perceptions of self-generated explanations provided by LLM-based Recommender System (LLM-Rec). We conducted a user study in which participants received movie recommendations and explanations generated by OpenAI's GPT-3.5 Turbo[4], the most advanced model available at the time, and rated the expla-

[3]https://platform.openai.com/docs/models/#gpt-4o

[4]https://platform.openai.com/docs/models/#gpt-3-5-turbo

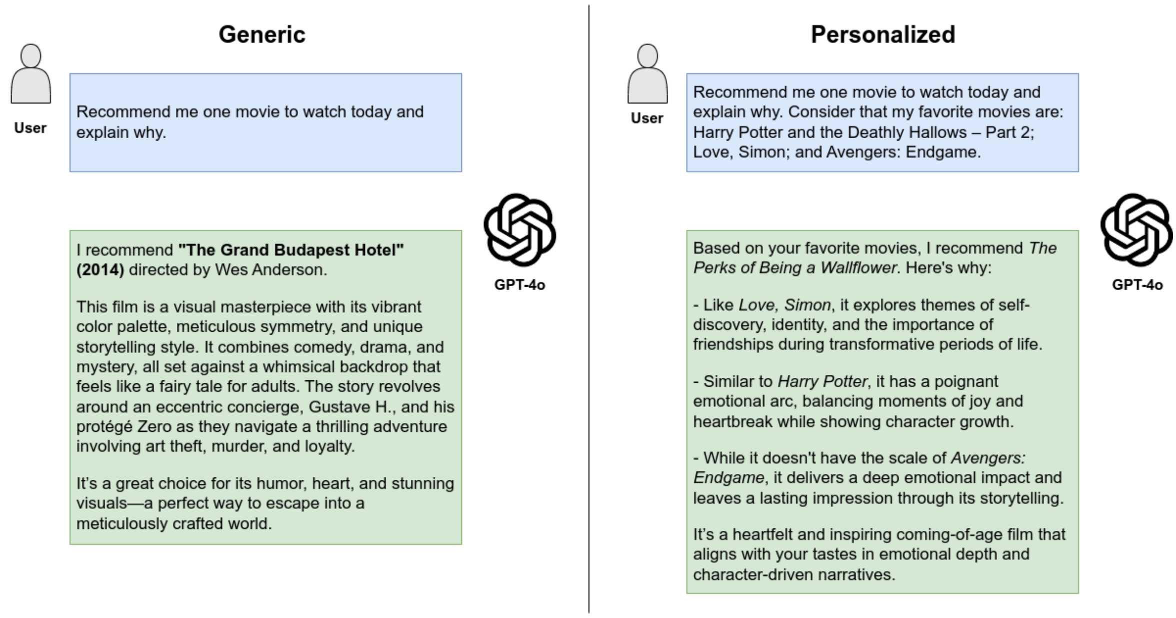


Figure 1.1: Example of interactions with ChatGPT for movie recommendation. The left chat depicts a generic recommendation/explanation (`https://chatgpt.com/share/678eacdc-5968-800f-945d-1debfb0d9b02`) and the right chat depicts a personalized recommendation/explanation (`https://chatgpt.com/share/678eae60-f928-800f-991a-df613cd57fee`).

a

nations on three dimensions: effectiveness, personalization, and persuasiveness (see **RQ1** to **RQ4** below). The movie domain was chosen because it is a popular area for recommendation tasks and an accessible domain where users are likely to have good prior knowledge.

Additionally, motivated by the highly persuasive tone observed in the explanations and the evident differences in the argumentation of generic and personalized explanations, we broke the explanations into sentences and used GPT-4o to classify them into types of arguments, aiming to understand how user-specific explanations compare to generic explanations regarding the types of argumentation they employ (see **RQ5** below).

Finally, we investigated whether the self-generated explanations reflected the model's actual rationale, i.e., were faithful or merely provided plausible justifications for the recommendations (see **RQ6** below). This topic is critical in the scenario of LLMs, as these models are susceptible to hallucinations and could, by providing plausible justifications, lead to misinformed decisions by the user, ultimately undermining the user's trust and the system's accountability. We performed an axiomatic evaluation based on the *Feature Agreement Axiom*, which posits that if the explanation is faithful, an important feature for a prediction or recommendation should be represented in the explanation with proportional importance. Since this experiment did not require an online evaluation, we expanded the scope to three domains—movies, songs, and books—and four models—GPT4o, Gemma 2 9B, Llama 3 70B, and Mixtral 8x7B.

In summary, we formulated six research questions (RQs) to guide these experiments, which are enumerated below:

RQ1: How do users value personalized ChatGPT-generated recommendations compared to random recommendations?

RQ2: How do users perceive user-based versus generic ChatGPT-generated explanations in relation to recommendation methods and explanation goals such as effectiveness, personalization, and persuasiveness?

RQ3: Do user-based versus generic explanations work differently for familiar or unfamiliar movies?

RQ4: How do explanation goals such as personalization, persuasiveness, and effectiveness relate?

RQ5: How do user-specific explanations compare to generic explanations regarding the types of argumentation they employ?

RQ6: Are the explanations provided by LLM-based Recommender Systems faithful?

Our results for **RQ1** to **RQ3** indicate that users prefer ChatGPT's personalized recommendations over random selections of popular movies. Surprisingly, even when ChatGPT bases its explanations on users' movie preferences, they are not perceived as more personalized than generic ones unless the recommendations are random. This insight also extends to the perceived effectiveness and persuasiveness of the explanations.

In **RQ4**, we further investigated the interconnectedness of personalization, persuasiveness, satisfaction, and explanation effectiveness. Our findings showed that explanation effectiveness is highly influenced by users' satisfaction, perceptions of persuasiveness, and personalization, with persuasiveness having the most significant impact. Moreover, for **RQ5**, we characterize the explanations by their argument types, considering 14 types included in the Periodic Table of Arguments [44] and find significant structural differences between generic and user-specific explanations.

Regarding the explanations' faithfulness and **RQ6**, the results revealed that LLMs do not always produce faithful explanations for recommendations, given that the explanations often failed to pass the axiomatic test. We found notable differences across models, domains, and recommendation types. GPT-4o outperformed other models in the evaluated metrics, especially in the movie domain. Positive recommendations generally yielded higher faithfulness scores, likely due to models' inherent bias toward avoiding negative content. Moreover, the movie domains led to better metrics, indicating that the LLMs' faithfulness depends on the domain. More studies are still necessary to understand the causes of these findings.

The remaining chapters of this thesis are organized as follows: Chapter 2 presents background about LLMs, Recommender Systems, and Explainable Artificial Intelligence (XAI), with a special focus on Faithfulness. Chapter 3 discuss the related literature. Chapter 4 and 5, respectively, detail the experiment design, results, and limitations regarding the **RQ1**-**RQ5** and **RQ6**. Lastly, Chapter 6 summarizes our findings and contributions.

Chapter 2

Theoretical Background

2.1 Large Language Models

The history of language models can be divided into several milestones, which took these models from traditional statistical methods to complex neural architectures. The early models represented words as numerical vectors, capturing semantic relationships using approaches such as Word2Vec [28] and GloVe [30]. However, these methods faced difficulties capturing long-range dependencies and contextual nuances within the text. The introduction of neural networks in the area brought considerable improvements and helped to deal with these limitations using Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) [15] networks, which enabled sequential processing of the data, thus allowing the models to retrieve the surrounding context of words in the text. However, RNNs struggled in performance due to their intrinsic lack of parallelization (requiring sequential processing) and the vanishing gradient problem. A key breakthrough in the field was the development of attention mechanisms [2], which allowed the models to focus on specific parts of the input, improving significantly the contextual understanding of the models. This mechanism led to the introduction of the Transformer architecture [41], which was based entirely on them. This architecture enabled massive parallelization by dropping the sequential dependency introduced by the RNNs and was able to capture complex relations within the text, forming the basis for most modern LLMs.

Building on the Transformer architecture, several prominent LLMs have emerged. Mixtral, developed by Mistral AI, uses Mixture-of-Experts (MoE) architecture, which mixes several small language models to enable high performance with fewer computational resources. From Google, the Gemma series [37, 38] is a family of open-source models built with the technology used in Google's closed-source model Gemini and has demonstrated good performance in several benchmarks for language understanding, reasoning, and safety. Created by Meta AI, the Llama family of models [12] is open-source and was used to build several others models due to its performance which is comparable to the OpenAI's closed-source state-of-art models. Finally, the GPT series of closed-source models from OpenAI [18], such as GPT-3 and GPT-4, have demonstrated impressive capabilities in text generation, translation, and question answering, being the current state-of-art family of models. Although sharing the transformer architecture, these models differ in scale, training methodologies, and specific application focus.

Although the training of these models focuses on predicting the next word of a sequence, LLMs demonstrated capabilities not explicitly programmed during training [5], such as classification of sentences, code generation, and recommendation of items of interest. They could do this with zero or a few examples within the input prompt (In-Context Learning (ICL)). They also exhibited the ability to perform chain-of-though reasoning (CoT) [47], which helped the models to produce more accurate solutions for complex problems by breaking them into smaller steps. These emergent capabilities highlight significant advances in language modeling, creating new opportunities and challenges for research in the field.

2.2 Large Language Models as Recommender Systems

Taking advantage of the emerging capabilities mentioned above, several research works have begun to evaluate their suitability for making recommendations. As this new research field started to flourish, paradigms about **WHERE** and **HOW** to use LLMs began to appear. The **WHERE** to use paradigm concerns which part of the recommendation pipeline will take advantage of the capabilities of these models. In contrast, the **HOW** to use intuitively reflects how these models are used.

Regarding the **WHERE** part, Lin et al. [20] identified works using these models over the entire recommendation pipeline. In the *feature engineering* step, for example, these models can be used to summarize user preferences from a profile or consumption history or as feature encoders by providing embeddings (numerical vectors) for textual inputs, which can be used for training traditional recommendation models. Another possible application is in *scoring and ranking*, i.e., these models can help generate the recommendations by receiving a set of recommendation options and user preferences. In turn, Wu et al. [50] categorized the usage in three categories: (1) *LLM embeddings + RS*, in which the role of the LLM is extracting embeddings from the raw input for being used by a traditional RS; (2) *LLM tokens + RS*, where the LLM is used to provide a summarization of the raw input as input for a traditional RS; and (3) *LLM as RS* where the model is used to provide the recommendations by receiving the input via prompting.

For the **HOW** to use paradigm, Zhao et al. [57] considered three approaches: (1) *pre-training*, where a new model is developed by learning the recommendation task; (2) *fine-tuning*, which involves taking a trained model and expanding its current knowledge to include the recommendation task; (3) *prompting*, where the recommendation is taught in runtime via ICL or CoT. Lin et al. [20] divided the approaches considering two aspects: whether or not to tune the model and whether the model is used to support a traditional RS or as RS itself.

The interaction of LLMs and recommender systems also brought new perspectives for personalizing the recommendations due to their enhanced language understanding and reasoning capabilities, as explored by Chen et al. [9]. In their work, they analyzed the different ways LLMs have been used for personalization and highlighted their usage as *knowledge basis*, *content interpreters*, *explainers*, *recommenders*, and *conversational agents*, for example.

However, the use of LLM for recommendation also comes with several limitations and new problems to solve regarding: *efficiency* (e.g., reducing training and inference time), *effectiveness* (e.g., mitigating hallucinations, dealing with long user consumption items, and user/item representations) and *ethics* (e.g., biases, fairness, safety, explainability, and privacy) [20, 50, 57]. Considering the scope of this work, the next section will dive deeper into the *ethics* considerations, especially those regarding explainability.

2.3 Explainable Artificial Intelligence

In recent years, AI has become ubiquitous, with millions of people interacting with some form of AI every day. These solutions have been applied to high-risk sectors such as health, security, transportation, and banking. To provide better results and more refined solutions, these models have grown in complexity and have become opaque even to their developers. This ubiquity, the difficulty in understanding the reasoning behind model predictions, and the need to comply with legal regulations promoted the emergence of an area of research called Explainable Artificial Intelligence (XAI) [1].

The XAI field aims to make AI models more comprehensible for humans by providing explanations for AI's decisions and predictions (*accountability/responsibility*) in a *fair/non-discriminatory* and *transparent* (e.g., detailing the model's reasoning) way [42]. Considering the scope of recommendations, an Explainable Recommender System aims to answer *why a particular recommendation was made*. Adadi and Berrada [1] summarized the motivations for XAI into four reasons that capture the essence of the need for XAIs models: (1) *to justify*, explain why a specific decision was made; (2) *to control*, explainable systems are easier to debug and fix; (3) *to improve*, explainable systems are more straightforward to improve because the developer can understand what needs improving; (4) *to discover*, models can discover new strategies to solve a problem, which may be yet not known by its users, if this model can explain its process, new knowledge is generated.

To address the need for explainability, researchers are delving into generating more transparent models and creating explanatory methods for models that cannot be interpreted naturally. The explanatory methods can be categorized considering some properties, such as: *scope*, explanations can be local when the objective is to explain a specific prediction or global when explaining the whole model behavior; *time*, whether the explanations are generated during (built-in) or after (post-hoc) the prediction; *model accessibility*, which parts of the model are accessible when generating the explanation; and others [25]. These methods can also be model *agnostic* when they work independently of the model architecture or *dependent* when they work only for specific models.

Considering the field of Natural Language Processing (NLP) and RS, which are the scopes of this work, most of the explanatory methods are based on techniques such as *feature attribution*, which tries to assign a relevance score for each feature in the input; *examples*, which tries to explain the prediction by providing similar (or dissimilar) predictions; *analysis of internal structures*, which aims to explain based on internal aspects of the models, such as neuron activations and attention weights; and *self-explanation*, especially in the case of LLM, where the models itself explains its decision [25, 56]. The evaluation of these methods can be done both by *user studies*, *online*, and *offline evaluations*, the latter being the easiest to carry out but less generalizable.

Lyu, Apidianaki, and Callison-Burch [25] summarized some principles that are desired for a model's prediction explanation: *faithfulness*, the explanation must reflect the inner reasoning of the model; *plausibility*, the explanation must be understandable by its target audience; *input/model sensitiveness*, changes in the input/model that affect (or not) the output should affect (or not) the explanation; *completeness*, the explanations must cover all the aspects that were relevant to a prediction; and *minimality*, the explanations should use the minimal set of aspects that explain the prediction. Although all these principles are important, given the scope of this research, we will explore the faithfulness principle in more detail.

An explanation for a prediction is faithful if it reflects the inner reasoning of the model for making that prediction [17]. These explanations are important because they can reveal causal relations between input and output, giving the user more transparency and information before accepting the prediction. Furthermore, unfaithful but still plausible explanations can be extremely dangerous depending on the domain, as they can convince the user to make a decision without knowing the risks [25].

Considering the scope of the LLMs and their ability to provide self-explanations, there is still no sufficiency test that can assure the faithfulness of these explanations, although several necessary tests have been proposed [29]. We can group the existing methods for evaluating the faithfulness of language model explanations into the following categories: *axiomatic evaluation*, applies necessary tests based on axioms; *predictive power evaluation*, assess whether the explanation for a given case can be used to predict accurately unseen cases; *robustness evaluation*, assess if similar outputs with similar inputs lead to similar explanations; *perturbation-based evaluation*, induces changes in the input and observe if the output also changes and if this change is reflected in the explanation; *white-box evaluation*, uses transparent models or transparent tasks; and *human-based evaluation*, uses humans to evaluate the explanations (this type of methods instead measures for plausibility since it assess if the model explanation matches human intuition) [25].

In this work, we used an *axiomatic evaluation* due to its simplicity, lack of reliance on ground truth, and applicability to black-box models. In contrast, *robustness evaluation* and *perturbation-based evaluation* would introduce additional challenges: the former requires identifying similar inputs with similar outputs, which may not always exist, while the latter relies on generating items with similar or dissimilar properties for perturbation. Similarly, *predictive power evaluation* would involve complex prompt engineering, including incorporating explanations into a prompt and determining which explanations to include.

2.4 Shapley Values

In game theory, a coalitional game is a cooperative game in which players form groups, or coalitions, to enhance their results in the game. Formally, a coalitional game G can be defined by the tuple (N, v), where N is a set of n players, and $v : 2^N \to \mathbb{R}$ is a characteristic function, with $v(\varnothing) = 0$, that returns the payoff obtained when a coalition $S \subseteq N$ plays the game G. Given a coalition game G, the ***Shapley Values*** are the unique way to fairly distribute the total payoff of a coalitional game while satisfying the following properties:

- **Efficiency**: The sum of the reward of all players equals the total payoff.
- **Symmetry**: Players with the same contributions receive the same reward.
- **Null player**: Players with no contributions receive no reward.
- **Linearity**: If two coalitional games are combined, a player's reward equals the sum of its reward in each game.

The Shapley Value of a player in a coalitional game, $\phi_i(N, v)$, represents their marginal contribution to the total payoff, indicating the proportion of the game's reward attributable to that player's participation [34]. It is computed by averaging the player marginal contribution for each coalition using the following equation:

$$\phi_i(N, v) = \frac{1}{n} \left(\sum_{S \subseteq N \setminus \{i\}} \frac{1}{C_{n-1}^{|S|}} \left[v(S \cup \{i\}) - v(S) \right] \right), \tag{2.1}$$

where $n = |N|$ is the number of players in N.

For each coalition S of a given size $|S|$, the term $v(S \cup \{i\}) - v(S)$ represents the marginal contribution of player i to the coalition S. Then, it is averaged by the total number of coalitions of size $|S|$, given by $C_{n-1}^{|S|}$. Lastly, there are n possible coalition sizes, i.e., $|S| = 0, 1, \ldots, n-1$, where $|S| = 0$ corresponds to $S = \emptyset$. So, to account for i's average marginal contribution across all coalition sizes, its Shapley value is averaged by n.

The ice cream example To illustrate the application of Shapley Values, we will explore a simple problem. Consider three kids: Ana, who has \$6; Bruno, who has \$4; and Charles, who has \$3. They want to buy ice cream, but none can afford it individually. The available ice cream sizes and their costs are 500g for \$7, 750g for \$9, and 1000g for \$11. The kids decide to pool their money together and then divide the amount of ice cream they can purchase among themselves. The Shapley Values represent a fair way to distribute the ice cream with respect to the properties mentioned above. Considering $v(S)$ as a function that indicates the amount of ice cream that the coalition S can purchase, to compute the Shapley Value for Ana, we analyze the coalitions excluding her ($S = \emptyset, \{B\}, \{C\}, \{B, C\}$):

1. For $|S| = 0$, we have $\frac{1}{C_2^0}[v(\{A\}) - v(\emptyset)] = 0$;
2. For $|S| = 1$, we have $\frac{1}{C_2^1}[v(\{A, B\}) - v(\{B\}) + v(\{A, C\}) - v(\{C\})] = \frac{1}{3}(750 + 750) = 750$;
3. For $|S| = 2$, we have $\frac{1}{C_2^2}[v(\{A, B, C\}) - v(\{B, C\})] = 1000 - 500 = 500$.

Thus, $\phi_A(N, v) = \frac{1}{3}(0 + 750 + 500) = \frac{1250}{3}$. By applying a similar process, we find $\phi_B(N, v) = \frac{875}{3}$ and $\phi_C(N, v) = \frac{875}{3}$. By these results, we can see that although having different amounts of money Bruno and Charles contributed the same (*symmetry*).

Application in Machine Learning Explainability The SHapley Additive exPlanations (SHAP) [23] method applies Shapley Values to explain the output of machine learning models in terms of their input features. By considering a coalitional game where the players are the features and the characteristic function is the model itself, Shapley values are computed by simulating scenarios in which specific feature values are considered present while others are treated as absent.

The method provides local explanations (i.e., at prediction level) in terms of feature importance with the guarantees that the prediction is fully explained (*efficiency*), features with similar importance receiving similar scores (*symmetry*) and features without importance receiving zero as a score (*null player*). However, given that computing Shapley Values has exponential cost, to be able to compute for a large number of features, the method uses only a sample of coalitions in the computation, which does not guarantee faithful Shapley Values. In this work, we also applied Shapley Values to compute input feature importance, as detailed in Section 5.1.

2.5 Mixed Effect Linear Models in Within-User Experiments

Mixed Effect Linear Models (MELMs) are statistical tools commonly used to analyze data that exhibit hierarchical or nested structures. They are particularly suitable for within-user experiments, in which individual users provide multiple responses across different conditions, leading to repeated measures data that require proper handling to account for both within-user and between-user variability [4, 6, 26].

MELMs incorporate two types of effects:

- **Fixed Effects:** Represent variables of interest consistent across the population, such as the experimental condition (e.g., the type of items being evaluated). These effects are used to estimate overall trends or differences that are generalizable.
- **Random Effects:** Account for variations specific to individual participants or other grouping factors. For within-user designs, a random intercept is typically included to model the baseline differences between users, and random slopes may be added to capture variability in how users respond to specific conditions.

The general form of a mixed effect model for within-user experiments is:

$$y_{ij} = \beta_0 + \beta_1 X_{ij} + u_i + \epsilon_{ij},$$

where:

- y_{ij} is the response for user i on item j.

- β_0 and β_1 are fixed effect coefficients.
- X_{ij} represents predictors (e.g., the conditions being tested).
- u_i represents the random effect for user i.
- ϵ_{ij} is the residual error term.

Teaching strategies example Consider that a school wants to investigate the impact of two new teaching strategies on students' performance: ***gamification*** and ***LLM-based tutoring***. Each student completed multiple assessments under the four possible combinations of these strategies (none, gamification with and without LLM-based tutoring, and LLM-based tutoring only). To understand the effect of these strategies individually and interactively on students' performance, we can fit a Mixed Effect Linear Model:

$$y_{ij} = \beta_0 + \beta_1 G_{ij} + \beta_2 T_{ij} + \beta_3 (G_{ij} \cdot T_{ij}) + u_i + \epsilon_{ij},$$

where:

- β_0 is the baseline score for assessments using traditional methods ($G = 0, T = 0$).
- β_1 is the fixed effect of gamification ($G = 1$) on performance.
- β_2 is the fixed effect of LLM-based tutoring ($T = 1$) on performance.
- β_3 is the fixed effect of the interaction between gamification and LLM-based tutoring ($G = 1, T = 1$).
- u_i is the random intercept for student i.
- ϵ_{ij} is the residual error term.

Consider that after fitting the model, we obtained $\beta_0 = 65$, $\beta_1 = 10$, $\beta_2 = 5$ and $\beta_3 = 3$ with significant effects. This allows us to conclude the following:

1. The average score under the traditional teaching was 65.

2. Individually gamification and the use of LLM-based tutoring contributed to increase the score in, respectively, 10 and 5 points.

3. Combining the gamification with LLM-based tutoring led to an increase of 3 points in the score, beyond the sum of their individual effects.

In our work, these models were used to evaluate the effect of different setups of recommendation conditions on users' perceptions of the recommended items and their explanations.

Chapter 3

Related Work

3.1 Explanations and Users' Perceptions

As detailed in Section 2.3, Explainable RSs have become an important research topic, with various methods proposed to generate explanations. These include leveraging the similarity of the recommended item with previously consumed items, assessing the item features' importance with respect to the user, and aggregating other users' reviews. Additionally, several presentation formats, such as textual, visual, and social explanations have been explored [53].

Chang, Harper, and Terveen [7] demonstrated the benefits of personalized natural language explanations over personalized tag-based explanations. They generated personalized natural language explanations by tasking crowd workers to create them by summarizing movie reviews related to the recommended item. Their solution was integrated within MovieLens, and it was perceived as more efficient and effective than the traditional tag-based explanations of the platform.

In a similar study, Lu et al. [22] highlighted the preference for human-generated explanations over machine-generated ones, such as those based on similarity to other users and consumed items, popularity, and content. These machine-generated explanations were presented in a textual format using template sentences. However, these solutions lack scalability, as using them for a large platform would require significant labor, leading to prohibitive costs.

The emergence of LLMs offers a promising solution to this scalability challenge. Models such as OpenAI's GPT-4[1], Meta's Llama[2], and Google's Gemma[3] have demonstrated remarkable capabilities in diverse tasks, including recommendation, as explored in Section 2.1 and Section 2.2. Several works have shown that these models can generate accurate and relevant recommendations, either independently or in combination with other methods, such as Knowledge Graphs [14, 16, 24, 35, 46].

Gao et al. [13] introduced Chat-REC, a conversational recommender system framework powered by ChatGPT. This system leverages users' profiles and consumption history to produce recommendations and explanations through natural language interaction with a user. Compared with traditional recommendation approaches, the results demonstrate ChatGPT's better performance in precision, recall, and nDCG metrics. However, no evaluation was performed regarding the quality of the explanations.

The evaluation of explanation quality has become increasingly important in recent years, mainly motivated by its impact on user trust in the system and for legal reasons, as explored in Section 2.3. Tintarev and Masthoff [39] introduced seven goals for an explanation: effectiveness, efficiency, persuasiveness, satisfaction, scrutiny, transparency, and trust. These goals have guided the dimensions of explanation quality evaluation in several works [3, 7].

In a study employing crowd workers to produce and evaluate personalized explanations for recommendations, Balog and Radlinski [3] found that these seven goals were often highly correlated, even when workers were instructed to produce explanations aimed at a specific goal. These goal-specific explanations were perceived as equally good compared to non-tailored ones, suggesting that either the crowd works lacked the ability to produce goal-specific explanations or that users could not distinguish between them.

Zhou and Joachims [58] presented one of the pioneering studies approaching the effect of personalized recommendations and explanations generated by the underlying LLM. In a survey with 120 participants, they compared the effectiveness of LLM-generated text reviews with human-generated ones. They found no significant differences in user perception, except for movies previously seen by the participants, in which LLM-generated reviews were favored.

In light of the aforementioned literature, **RQ1** (satisfaction with personalized vs. random

[1]https://openai.com/index/gpt-4/

[2]https://www.llama.com/

[3]https://ai.google.dev/gemma

recommendations) is settled on the works of Harrison, Dereventsov, and Bibin [14], Huang et al. [16], and Gao et al. [13], which demonstrate the great potential of GPT models in generating accurate and relevant recommendations based on user information. We aim to assess whether the users experience personalization, a gap left by these works. This question validates whether the GPT models can provide a discernibly personalized experience compared to random recommendations.

For **RQ2** (perception of user-based vs. generic explanations), we were inspired by the findings of Balog and Radlinski [3], who showed that personalized explanations were perceived as equally pleasing to non-personalized ones. This perception goes against common sense and highlights the importance of understating how users perceive *user-based* versus *generic* explanations concerning different explanation goals.

On the other hand, **RQ3** (explanations for familiar vs. unfamiliar movies) examines whether familiarity with the recommendation affects user perception of the explanations, in line with the results of Zhou and Joachims [58], who observed differing experiences regarding familiar and unfamiliar movies.

In **RQ4** (relationship between explanation goals), we aim to explore the interactions between different explanation goals. This question arises from the results of Balog and Radlinski [3], in which the goals were highly correlated. However, in its study, these relations were not investigated in depth, so in our study, we examine the extent to which the effectiveness of the explanation and satisfaction with the recommendation relies on personalization and persuasiveness.

Lastly, **RQ5** (argumentation in user-based vs. generic explanations) expands upon the research by Balog and Radlinski [3] and adds to **RQ2** by examining the tone of discourse in ChatGPT's explanatory outcomes. It investigates how differences in the model's understanding of users' preferences affect the types of arguments it employs.

In summary, these research questions, **RQ1** to **RQ5**, were designed to build upon one another, providing a sequential understanding of the users' perceptions of LLM-Recs in producing personalized explanations for recommendations.

3.2 Faithfulness for LLM-generated explanations

Jacovi and Goldberg [17] state that faithful explanations accurately represent the model's reasoning behind its predictions. Although LLMs can generate self-explanations for their tasks (including recommendations), they remain black boxes and produce plausible but unfaithful explanations [40, 52]. In recent years, significant efforts have been made to explain the rationale behind these generative models. The remainder of this section details research works that inspired the experiment design for answering **RQ6**.

In their work, Zhao et al. [56] reviewed the literature on explainability for LLMs and introduced a taxonomy of techniques, classifying them into two main paradigms: fine-tuning-based and prompt-based. The fine-tuning paradigm focuses on understanding how fine-tuning processes contribute to the model's improved performance on new tasks. In contrast, the prompt paradigm seeks to explain how the model leverages its pre-trained knowledge to respond to prompts across a wide range of tasks. Within the prompting paradigm, input and label perturbation emerged as the most common methods for explainability. These techniques involve modifying the model's input and examining the resulting changes in output. The paper also emphasizes the importance of generating faithful explanations and calls for further research into innovative techniques that accurately capture the model's reasoning processes.

Parcalabescu and Frank [29] evaluated various tests proposed in the literature to measure faithfulness across eleven versions of open-sourced LLMs (Llama2, Mistral, Falcon, GPT2) on five tasks. They noted that most current faithfulness tests primarily assess self-consistency rather than faithfulness. Self-consistency refers to a model's ability to produce consistent outputs and explanations when inputs are varied. However, because the internal mechanisms of LLMs for prediction and explanation may differ, self-consistency alone does not ensure faithfulness. They also introduced a novel self-consistency test called CC-SHAP, which uses the output probabilities to compute Shapley values to quantify input tokens' influence on prediction and explanation and then compare them.

Their test assumes that faithful explanations should exhibit a similar distribution of input tokens' Shapley Values for both the output and the explanation, i.e. the important input tokens must be the same for the output and explanation generation. Unlike other tests, CC-SHAP provides a continuous measurement of faithfulness rather than a binary one and improves explainability by offering a distribution of token-level importance. As detailed in Section 5.1, we also compare the importance of the input for both output and explanation using Shapley Values for obtaining the input items' importance in relation to the output. However, CC-SHAP was not applicable since it requires access to the output probabilities, which are unavailable for closed-sourced models, like OpenAI's GPT models.

Matton, Ness, and Kiciman [27] introduced the concept of *causal concept faithfulness*, considering that when explaining, the LLMs refer to high-level concepts implicitly present in the input rather than specific words or tokens. Their approach involves five steps: (1) extracting a complete set of semantic concepts in the input using GPT-4, (2) generating perturbed questions to identify causal effects through counterfactual reasoning, (3) estimating the actual causal effects of concepts using a logistic regression model, (4) identifying the subset of concepts implicated in the model's explanations, and (5) computing faithfulness metrics, such as false reference rate and omission severity, based on discrepancies between the actual causal set and the concepts used in the explanations.

They proposed three metrics: *False References Rate*, which measures the fraction of concepts mentioned in the explanation that does not have a causal effect; *Omission Rate*, which measures the fraction of concepts with a causal effect that is not mentioned in the explanations; and *Omission Severity Rate*, which quantifies the impact of the omitted concepts using a utility function. We draw inspiration from these metrics to compute *Recall* and *Precision*, as detailed in Section 5.1.

In summary, we agree with the literature on the innate challenges of assessing the *faithfulness* of LLM-generated explanations. Consequently, we adopted an *axiomatic evaluation* as a practical approach, treating it as a necessary but not sufficient condition for faithfulness. While this is not an ideal measure, it remains valuable by aligning with the self-consistency perspective outlined by Parcalabescu and Frank [29].

Chapter 4

Leveraging ChatGPT for Automated Human-centered Explanations in Recommender Systems

This chapter details the experiment to answer research questions **RQ1** to **RQ5**. This experiment involved a user study with 94 participants recruited using Prolific[1], a crowd-sourced platform often used for research with online experiments. The study occurred across three batches between June 8-15, 2023. First, each participant was required to inform their movie preferences by providing the names of three liked and three disliked movies. Next, we presented the participant with four movie recommendations and two movie disrecommendations accompanied with an LLM-generated natural language explanation of why the user should (or not) watch the movie. Finally, we collected the users' perceptions of recommendations and explanations using a five-question questionnaire and analyzed them using multilevel mixed linear regression. These steps are detailed in Section 4.1, followed by the presentation of the results in Section 4.2 and a discussion of the experiment limitations in Section 4.3.1.

[1]https://www.prolific.com

4.1 Experiment Design

4.1.1 Collecting user preferences

In order to collect user preferences and their evaluation of the recommendations and explanations, we developed a web application. This application comprises a frontend in React[2] and a backend with a REST API in Python[3].

On entering the website, the users were asked to insert three liked and three disliked movies, as shown in Figure 4.1. They could search and then choose a movie using a search bar powered by OMDb API[4]. Due to the GPT-3.5 knowledge cutoff being September 2021, only movies released before 2021 were shown as results in the search.

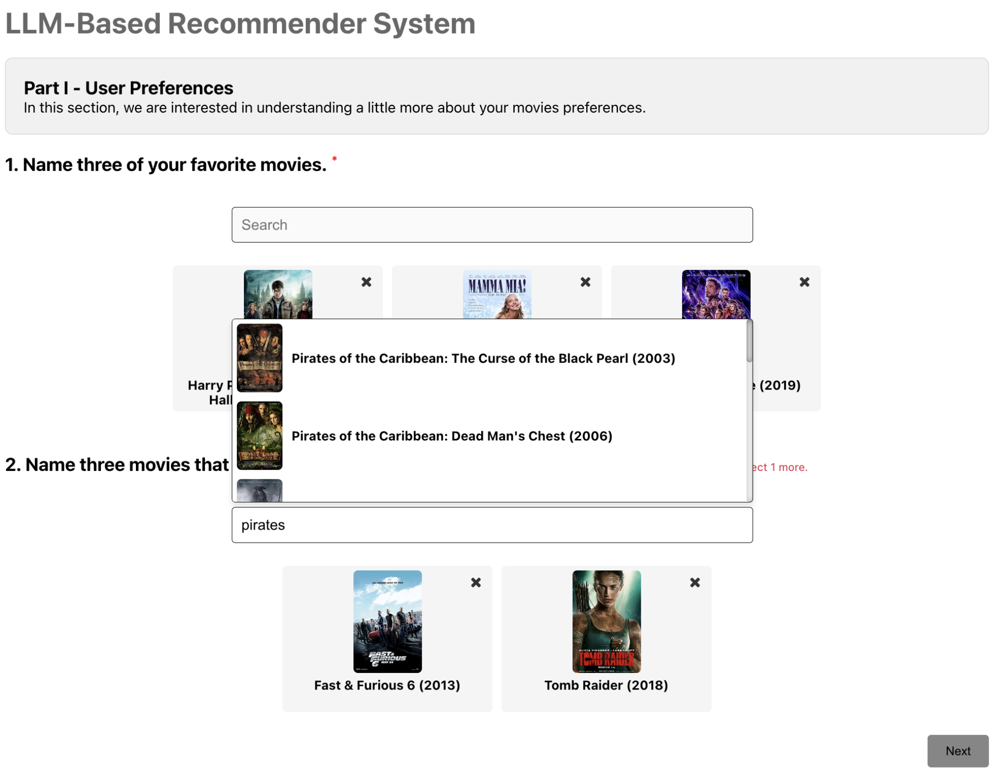


Figure 4.1: Overview of user preferences elicitation. The user was asked three movies they liked (*Name three of your favorite movies.*) and three movies they disliked (*Name three movies that you really disliked (or hated).*). Here, the user answers the question about the liked movies and searches for *pirates* in the search bar of the disliked movies.

[2]https://react.dev/
[3]https://www.python.org/
[4]https://www.omdbapi.com

Once the users had finished filling out the required information, their preferences were sent to the backend to obtain the recommendations and their respective explanations. The details of how this step was performed are presented in the following section.

4.1.2 Generating recommendations and explanations

Using the information provided on the user preferences, we were able to generate four recommendations and two disrecommendations, all accompanied by an explanation. A disrecommendation can be interpreted as a negative recommendation, i.e., a movie the user should avoid since it may not relate to their preferences. These negative recommendations might be equally valuable to users, as avoiding a potentially poor movie experience can be as beneficial as finding a good one. By showing positive and negative recommendations, we provide a wider range of feedback opportunities, allowing us to compare if user perception varies according to the intention of the recommended item [54].

To obtain the (dis)recommendations, we used two different Recommender Systems. In total, each user received four recommendations and two disrecommendations. Two of the recommendation and the two disrecommendations were generated by the LLM-Rec, which was built using the GPT-3.5 Turbo[5], an OpenAI model. By the time of the experiment, this was the best model publicly available from OpenAI[6]. Since this model is closed-sourced, we accessed it through Python using OpenAI's API for Chat[7].

The random recommender provided the remaining two recommendations by randomly selecting from a collection of popular films listed on IMDb[8]. Randomly recommending movies helps control for user familiarity by increasing the chances of recommending an unfamiliar movie. Recommending only movies users already know could bias their perceptions due to prior knowledge or experiences. By introducing unfamiliar movies through random recommendations, the research can better isolate the effect of the explanation itself on user decisions, regardless of prior familiarity with the movies.

The GPT-3.5 Turbo model is a generative natural language model, so to use it as an LLM-

[5]https://platform.openai.com/docs/models#gpt-3-5-turbo

[6]We used a specific snapshot of the model (*gpt-3.5-turbo-0613*) during the experiments, which is no longer available.

[7]https://platform.openai.com/docs/api-reference/chat

[8]https://www.imdb.com/chart/moviemeter/

Rec, we needed to create natural language prompts that could serve as input for the model. We adopted a zero-shot approach, providing the model solely with the task and input without any examples illustrating how to execute the task. Figure 4.2 depicts each prompt we used.

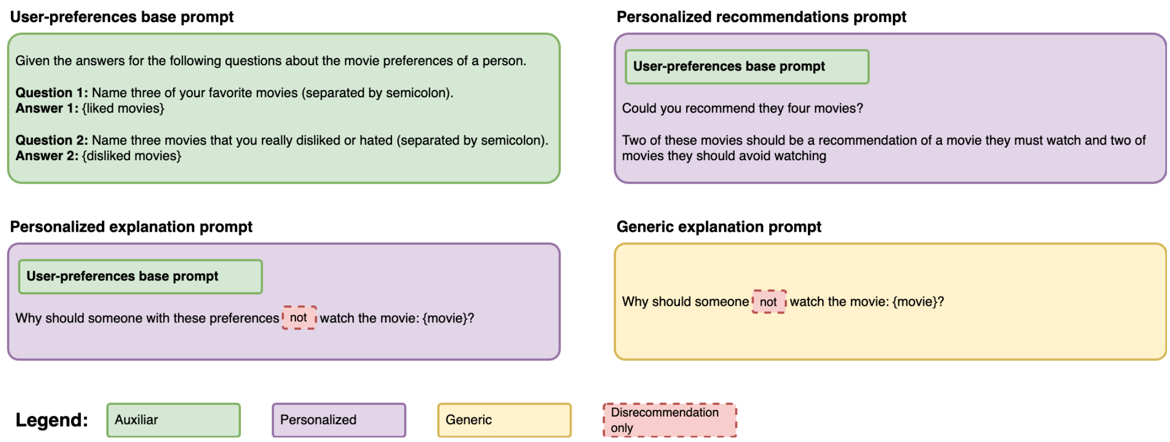


Figure 4.2: Prompts used for generating recommendations and explanation from the OpenAI GPT3.5-Turbo model

We created a base prompt to keep the user preferences (upper left). This prompt included the answers for two questions, respectively, *Name three of your favorite movies* and *Name three movies that you really disliked or hated*. The answers to these questions were collected as detailed in Section 4.1.1. This prompt was concatenated in each prompt that required personalization. Next, we had a prompt requiring the model to provide four movie recommendations according to the user preference (upper right). Two of these recommendations are must-watch movies, while the other two are movies the user should avoid.

Lastly, we wrote two prompts to obtain the explanations for the recommendations. The first (lower left) asked the model to provide a personalized explanation and included information about the user preferences. The second one (lower right) asks for a generic explanation, not having access to any user information. In both cases, a **not** was added when dealing with a disrecommendation.

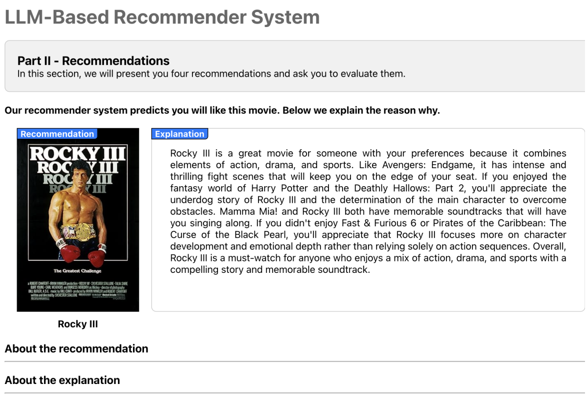
The prompts were crafted according to the best practices for In-Context Learning (ICL)[9]. To enhance usability, we integrated additional instructions regarding the expected answer format within each prompt. Moreover, we set the temperature parameter to its minimum value of 0 to promote reproducibility. This adjustment aimed to heighten the model's predictability

[9]https://huggingface.co/docs/transformers/main/tasks/prompting

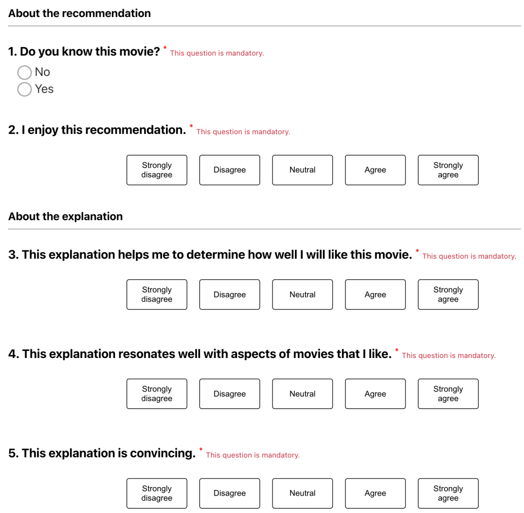
and coherence.

4.1.3 Collecting user perceptions

After receiving the response from the backend, we presented the user with the movies and explanations (one at a time). We asked them to answer a questionnaire with one yes-or-no and four Likert-scale questions, as depicted in Figure 4.3. The Likert-scale questions were affirmations, which the user should answer with one of the following options: *Strongly disagree*, *Disagree*, *Neutral*, *Agree*, *Strongly agree*.



(a) Evaluation page presented to the user



(b) Questionnaire

Figure 4.3: Overview of the evaluation step.

The questionnaire included two questions about the recommendation and three about the explanation. The first question, namely "*Do you know this movie?*", aimed to obtain information about the user's familiarity with the recommendation (aligned with **RQ3**). Meanwhile, the second question ("*I enjoy this recommendation*") was aligned with the **RQ1** and allowed us to compare how much the user liked the recommendations provided by the *LLM* and the *Random Recommender*.

The last three questions aimed to cover the aspects of the explanation we intended to measure for **RQ2**, respectively, effectiveness ("*This explanation helps me to determine how well I will like this movie*"), personalization ("*This explanation resonates well with the as-*

pects of movies that I like"), and persuasiveness ("*This explanation is convincing*"). These answers also allowed us to perform the correlation analysis for **RQ4**.

To analyze the answers, we employed multilevel mixed linear regression [33] with a random intercept model, accounting for the repeated nature of participant responses. Section 4.2 presents the experiment results.

4.2 Results & Discussion

4.2.1 Recommendation Satisfaction

Regarding **RQ1**, we analyzed the satisfaction experienced by the participants with a recommended movie considering two factors: the perception of personalization of the recommendations and their familiarity with the film. Both models yielded a significant amount of unfamiliar movies. According to the answers, 25% of the LLM-Rec's recommendations and 49% of the random recommendations were unknown to the users. Our results indicated that participants enjoyed random recommendations less than those personalized by GPT-3.5 ($\beta = -0.53, p < 0.001$). Moreover, the participants also rated unfamiliar recommendations less enjoyable than familiar ones ($\beta = -0.78, p < 0.001$). These effects are depicted in Figure 4.4a. No interaction effect was found between familiarity and recommendation method, which suggests that LLM-Rec recommendations are often preferred over random (yet popular) ones, regardless of the movie's familiarity.

4.2.2 User-based vs. Generic Explanations

For **RQ2**, we compared personalized and generic explanations for three explanation goals: effectiveness, personalization, and persuasiveness. Our results indicated that explanations provided by the GPT-based recommender were identified as more effective than those generated for random recommendations ($\beta = 0.37, p < 0.001$). However, we did not find any significant differences in the effectiveness of user-based versus generic explanations ($\beta = -0.10, p = 0.37$), as shown in Figure 4.4b.

In order to determine the perception of personalization, the participants were asked if the explanations aligned with their movie preferences. Surprisingly, the user-based explanations,

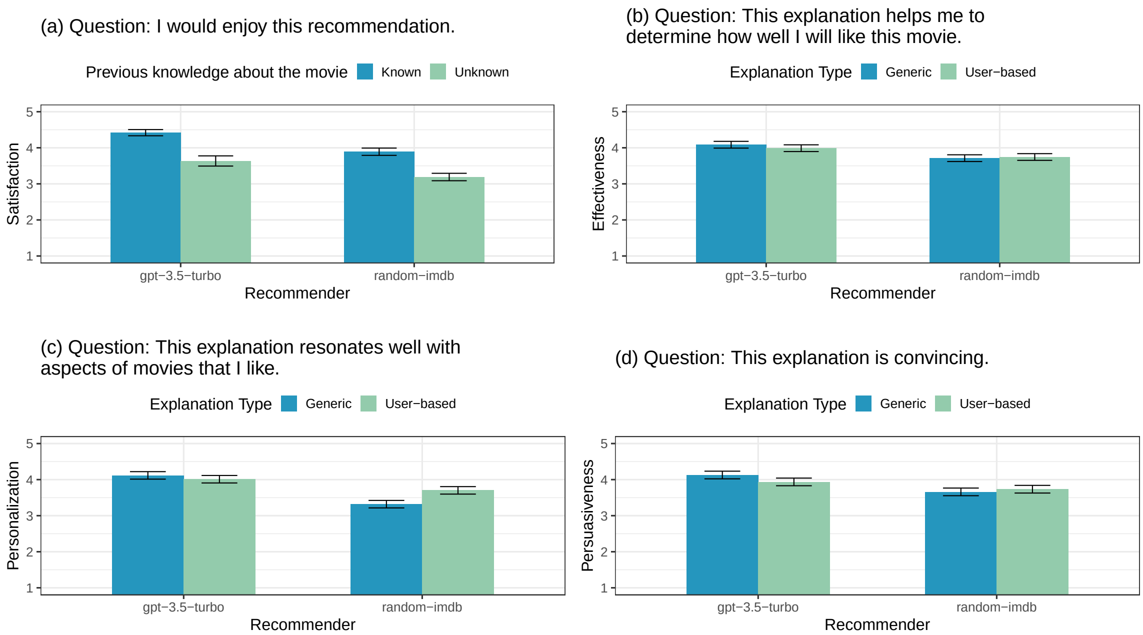


Figure 4.4: Questionnaire results for the (four) recommendations based on estimated means from the random intercept multilevel regressions. Responses were given on a 5-point, disagree-agree scale, for which 3 means neutral. Error bars are one standard error of the mean.

which mainly referenced participants' preferences, were not perceived as significantly more personalized than generic ones, except for random recommendations, which typically score lower on personalization ($\beta = -0.80, p < 0.001$). This result is illustrated in Figure 4.4c and supported by a significant interaction between recommendation's and explanation's type in our model ($\beta = 0.49, p < 0.01$).

Regarding persuasiveness, we found that the explanations are less convincing for random recommendations ($\beta = -0.47, p < 0.001$), with no significant difference between user-based and generic explanations. For the LLM-Rec recommendations, personalized explanations appeared slightly less persuasive than generic ones, as shown in Figure 4.4d. However, this difference was not statistically significant ($\beta = -0.27, p = 0.14$).

4.2.3 Movie Familiarity Analysis

For **RQ3** (cf. Figure 4.5), we investigated whether perceptions of effectiveness, personalization, and persuasiveness of explanations differed between familiar and unfamiliar movies.

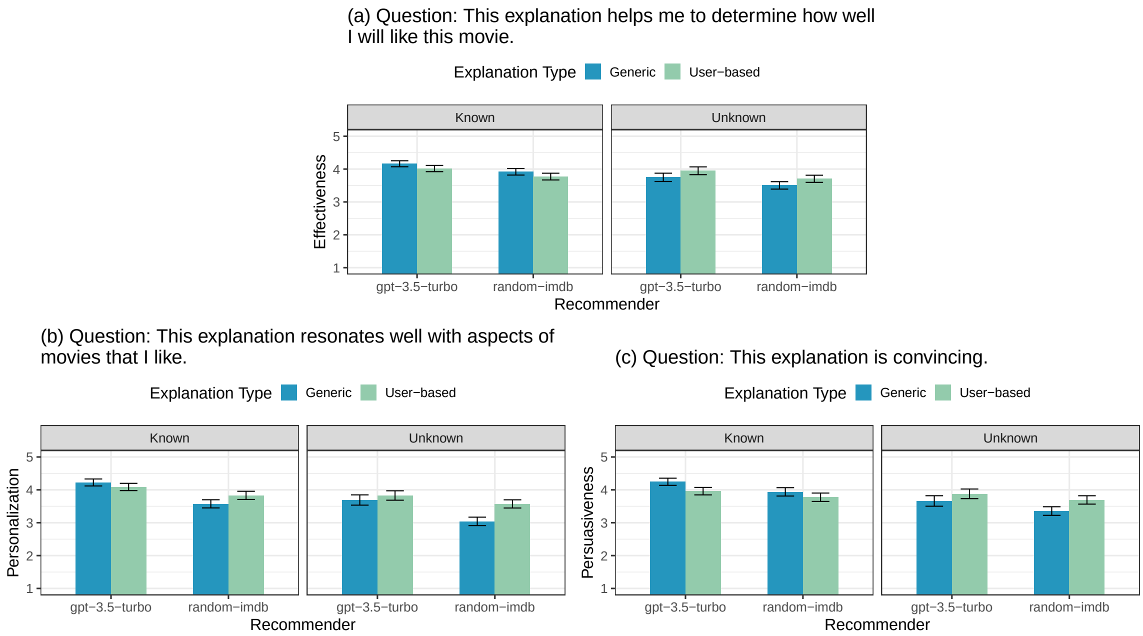


Figure 4.5: Results regarding how movie familiarity affects explanations' effectiveness, personalization, and persuasiveness.

Regarding effectiveness, explanations for unfamiliar movies were generally rated as less effective ($\beta = -0.41, p < 0.01$), primarily for generic explanations, as shown in Figure 4.5a. User-based explanations, however, maintained consistent effectiveness in both conditions, particularly benefiting unfamiliar movies, which is reflected in the positive interaction between explanation type and familiarity ($\beta = 0.35, p < 0.05$).

For personalization, there was a main effect of familiarity, with explanations for unfamiliar movies feeling less personalized overall ($\beta = -0.53, p < 0.001$), as illustrated in Figure 4.5b. Although user-based explanations appeared less influenced by familiarity, the interaction was not significant ($\beta = 0.274, p = 0.17$).

Concerning persuasiveness, explanations for unknown movies were perceived as less convincing overall ($\beta = -0.58, p < 0.05$). However, we noticed a significant main effect of explanation type ($\beta = -.28, p < .05$), interacting with familiarity ($\beta = .50, p < .05$). As shown in Figure 4.5c, user-based explanations were somewhat less persuasive for familiar movies but more persuasive for unfamiliar ones.

These findings suggest that user-based explanations outperform generic ones, particularly for unfamiliar movies. When users lack prior knowledge about a movie, user-based explanations that connect the movie to their preferences are more impactful than when they are already familiar with it.

4.2.4 Path Modeling of Explanation Types and Goals

In **RQ4**, we explored the relation between the different users' perceptions and experiences concerning personalization, persuasiveness, and effectiveness. We fitted a path model and used it to predict to what extent the explanation effectiveness is influenced by the reported levels of personalization, persuasiveness, and satisfaction with the movie and its explanation. Following the user-centric framework of Knijnenburg et al. [19], we see personalization and persuasiveness as perceptions (Subjective System Aspects: SSA), whereas satisfaction and effectiveness are experience-type constructs (EXP). Our conditions are the objective system aspects (OSAs) that affect SSAs and EXPs.

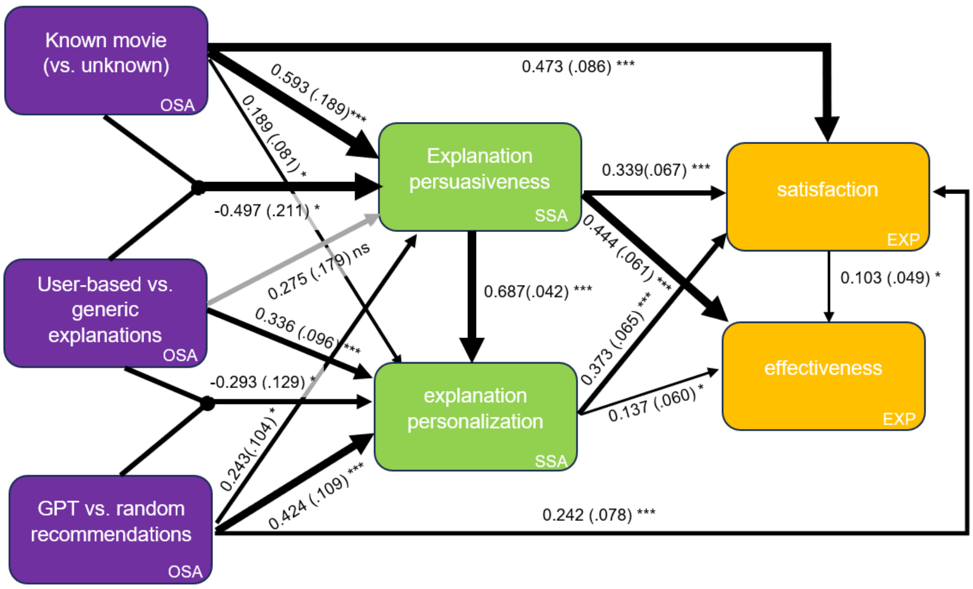


Figure 4.6: Path model showing how persuasiveness and personalization of the explanations are affected by the conditions and how they subsequently predict satisfaction and effectiveness. OSA=Objective System Aspect, SSA=Subjective System Aspect, EXP=Experience. The thickness of the line represents the strength of the coefficient. Standard errors in brackets, significance: * p<.05, **, p<.01, ***, p<.001

Aligned with the results of Balog and Radlinski [3], we identified a correlation between persuasiveness, personalization, satisfaction, and effectiveness, which we explore in depth in our path model. We found that effectiveness is predicted by satisfaction, persuasiveness, and personalization, with persuasiveness being the strongest predictor. Satisfaction itself increases with familiar movies and LLM-Rec recommendations (as previously demonstrated in the **RQ1** analysis) but is also directly influenced by persuasiveness and personalization.

Befitting our **RQ2** analysis, personalization was shown to be influenced by the type of explanation, type of recommendation, their interaction, and by the movie familiarity, with the last having a less significant effect (refer to Figure 4.4b that shows the same patterns), as well as directly by the level of persuasiveness[10]. Also, similar to the previous analysis for **RQ2** and **RQ3**, we found that persuasiveness is affected by the explanation type and movie familiarity (and their interaction) as well as by the type of recommendation.

Therefore, the path model confirms the effects discussed in the previous sections while providing insights into their relationships. An effective explanation is personalized, persuasive, and ideal for a satisfactory movie. Persuasiveness is influenced by explanation type, recommendation type, and movie familiarity, while personalization depends on explanation type, recommendation type, and persuasiveness. These findings clarify the conditions under which an explanation becomes effective and how to achieve this.

4.2.5 Disrecommendation Analysis

Our analysis also covered the two disrecommendations. Those were always generated by the LLM-Rec since a random recommendation provides no guarantees that we are recommending movies the user may not like. So, here, we do not consider the effect of the recommendation type but only the effects of explanation type and familiarity.

We found that generic explanations seemed more effective than personalized ones ($\beta = -.32, p < .05$), with no significative interaction with familiarity, as can be seen in Figure 4.7a. We could not find a significant effect for personalization for either of the two dimensions. For persuasiveness, we found that user-specific explanations are, to some extent, less convincing ($\beta = -.36, p < 0.05$). In general, explanations for disrecommendations seemed to not benefit from personalized explanations. However, further analysis is required to identify whether this effect has not originated from ChatGPT's reluctance to disrecom-

[10]It could be argued that persuasiveness increases with personalization; however, modeling this relationship resulted in a reduced fit.

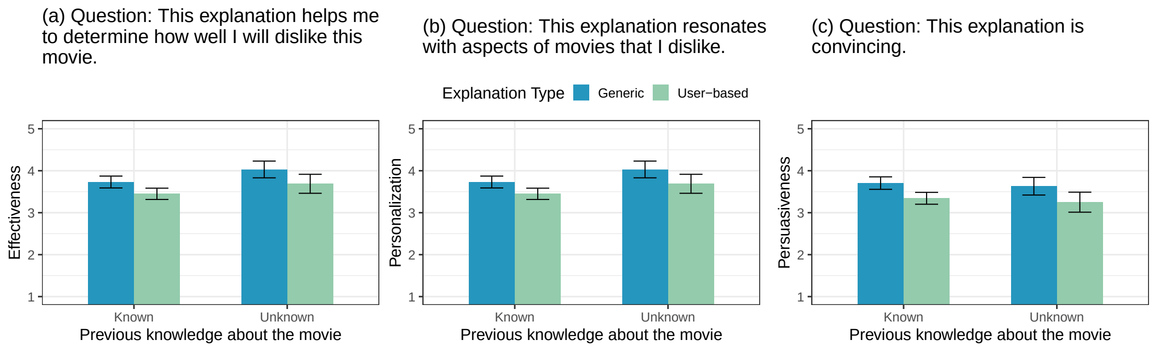


Figure 4.7: Questionnaire results for the two disrecommendations, based on estimated means from the random intercept multilevel regressions. Responses were given on a 5-point, disagree-agree scale, for which 3 means neutral. Error bars are one standard error of the mean.

mend, given OpenAI's efforts to avoid harmful and biased discussions [11].

4.2.6 The Anatomy of Explanations and its Influence on the User's Perception

Our research indicates that generic explanations can be seen as effective, persuasive, and personalized as those tailored to individual users, as noted by Balog and Radlinski [3]. However, individuals favor user-specific explanations, especially when faced with unfamiliar or random recommendations. To understand the underlying reasons for these observations, we analyzed the explanations into fundamental narrative components—specifically, the arguments. Using GPT-4o[12], we classified sentences in the explanations according to their argument types, intending to uncover patterns that could clarify user preferences.

We selected GPT-4o for this task because there was no suitable labeled dataset available for training a traditional supervised model. The prompt we used in our method is described in Prompt A.1.1. Sentences were categorized as either non-argumentative or as one of 14 different types of arguments derived from the Periodic Table of Arguments (PTA). The PTA serves as an extensive framework that systematically categorizes types of argumentation based on principles of classical dialectic (philosophical reasoning) and rhetoric (persuasive communi-

[11] https://openai.com/policies/usage-policies
[12] `https://platform.openai.com/docs/models#gpt-4o`

cation). It offers a structured approach to analyze argument schemes, fallacies, and methods of persuasion by grouping them into specific quadrants [44, 45].

This structure makes the PTA particularly appropriate for our research, allowing for a detailed breakdown of explanatory constructs into fundamental narrative components. Table 4.1 provides a ranked list of the identified argument types, organized by their frequency of occurrence. For each type, it also presents definitions and an example sentence from the dataset that demonstrates its application.

Table 4.1: Types of arguments present in the explanations

Type of Argument	**Definition**	**Frequency (%)**	**Example**
From Evaluation	Argues based on personal experiences of good/bad or effective/ineffective.	44.39	I would not recommend watching Transformers: Age of Extinction because it received negative reviews from both critics and audiences.
No Argument		22.93	Divergent is set in a dystopian society where people are divided into factions based on their personality traits.
From Similarity	Infers a fact based on its similarity to another fact.	9.46	The movie has a similar tone to Saw, which was listed as one of the person's favorite movies.
From Comparison	Evaluates a situation, object, or idea by comparing it to another.	8.00	Additionally, the movie has been criticized for its weak plot and character development, which may not be satisfying for someone who enjoyed movies like Bridesmaids.

Table 4.1: Types of arguments present in the explanations *(continued)*

Type of Argument	**Definition**	**Frequency (%)**	**Example**
From Criterion	Makes an argument considering specific criteria.	6.13	Additionally, the dislike for action movies like Need for Speed and superhero movies like Captain Marvel and Avengers: Endgame suggests that the person may prefer movies that are more grounded in reality and have a more serious tone.
Pragmatic	Evaluates actions, events, or rules based on their favorable or unfavorable consequences.	2.42	Save yourself the time and watch something else instead.
Ad Populum	Asserts something is true or correct because many people think so.	1.94	The movie has been praised for its intricate plot, beautiful cinematography, and strong performances by the cast.
From Opposites	Concludes something by presenting the persuasiveness of its opposite.	1.65	While Sing is not a dystopian film like Sucker Punch, it does have a message of perseverance and following your dreams, which may resonate with someone who enjoys that theme.

Table 4.1: Types of arguments present in the explanations *(continued)*

Type of Argument	Definition	Frequency (%)	Example
From Authority	Relies on the opinion of an authority figure as evidence.	0.95	The movie has received critical acclaim for its performances, direction, and cinematography, and has won several awards, including six Oscars.
From Sign	Asserts that the presence or absence of one thing indicates the presence or absence of another.	0.66	Finally, the person's love for Bohemian Rhapsody may indicate a preference for movies with a strong musical element, but Cats' unique style of music may not be to their liking.
From Analogy	Argues that because two things are similar, what is true of one is also true of the other.	0.59	Although it may not seem like it has much in common with the movies you listed, there are a few reasons why you might enjoy it.
From Commitment	States a claim supported by something the addressee has previously said.	0.40	If you're a fan of the Star Wars franchise, then you definitely shouldn't miss The Empire Strikes Back.
From Effect	Draws from consequences to infer the cause.	0.26	Therefore, it is unlikely that someone with these movie preferences would enjoy The Emoji Movie.

Table 4.1: Types of arguments present in the explanations *(continued)*

Type of Argument	**Definition**	**Frequency (%)**	**Example**
From Disjuncts	Concludes something happened because an alternative did not.	0.18	Since the person did not like the first movie, it is unlikely that they will enjoy the sequel.
From Equality	Based on the principle of equality, asserting that similar circumstances should lead to similar treatment.	0.04	The film's themes of loyalty, betrayal, and the corrupting influence of power are as relevant today as they were when the movie was first released in 1972.

The *Argument from Evaluation* was the most common type, comprising 44.39% of the sentences. This argument is structured as *a is X because a is Y*, where the premise presents a *statement of value* and the conclusion suggests a *statement of policy* (i.e., an action). For example, as shown in Table Table 4.1, the premise is ***Transformers: Age of Extinction received negative reviews from both critics and audiences***, while the conclusion is ***I would not recommend watching Transformers: Age of Extinction***, which fits this categorization.

The second most common classification is *No Argument*, covering 22.93%, indicating that nearly a quarter of the sentences do not present an argument but only a fact about the movie. Additionally, arguments from *similarity* (9.46%) and *comparison* (8%) also had a significant frequency, primarily due to the user-based explanations, as depicted in Figure 4.8. Both arguments are of the form: *a is X, because b is X*, distinguished by the types of statements in the premise and conclusion.

For the *similarity argument*, the premise and conclusion are statements of fact. Meanwhile, both are policy statements for the *comparison argument*. In other words, *arguments from similarity* tries to infer a fact based on its similarity to another fact, and the *argument from comparison* evaluates a situation, object, or idea by comparing it with another.

Figure 4.8 depicts the distribution of the classifications considering the two types of ex-

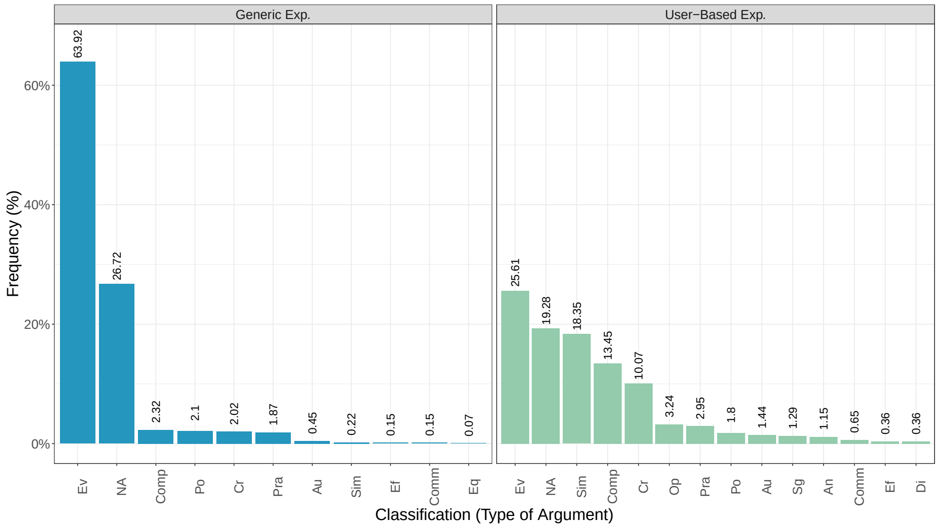


Figure 4.8: Distribution of the types of arguments grouped by the type of explanation. Classifications codes: NA (No Argument), An (Argument from Analogy), Au (Argument from Authority), Comm (Argument from Commitment), Comp (Argument from Comparison), Cr (Argument from Criterion), Di (Argument from Disjuncts), Ef (Argument from Effect), Eq (Argument from Equality), Ev (Argument from Evaluation), Op (Argument from Opposites), Sg (Argument from Sign), Sim (Argument from Similarity), Po (Argumentum ad Populum), Pra (Pragmatic Argument).

planations. *Arguments from evaluation* are the most common in both kinds of explanations. However, their prevalence differs. For Generic Explanations, they represent over half of the argumentative sentences. On the other hand, in User-based Explanations, the distribution is more evenly split among *arguments from evaluation*, *comparison*, and *similarity*. This variation is anticipated, as the preferences outlined in the prompt for User-based Explanations encourage the use of comparative arguments. We did not find any notable differences in the distributions concerning other factors, such as the recommender, the type of recommendation (positive or negative), and the placement of the sentence within the explanation.

Figure 4.9 shows examples of user-specific versus generic explanations produced by the GPT3.5-Turbo model for three different types of recommendations (GPT-generated recommendation, GPT-generated disrecommendation, and random recommendation). In line with the arguments' classification, we identified four categories utilized by the model to create the explanations.

GPT Recommendation

Personalized Explanation

The Shawshank Redemption is a classic movie that tells the story of a man who is wrongly convicted of murder and sent to prison. It's a movie about hope, friendship, and the power of the human spirit. If **you** enjoyed the themes of friendship and loyalty in Harry Potter and the Goblet of Fire, the mind-bending plot of Inception, and the sports documentary Johan Cruijff-En un momento dado, then **you** will definitely appreciate the depth and complexity of The Shawshank Redemption. On the other hand, if **you** disliked the superhero movie Doctor Strange, the action-packed IMAX Presents: Ant-Man and the Wasp, and the romantic drama Titanic, then **you** may find The Shawshank Redemption to be a refreshing change of pace. The movie has a 9.3 rating on IMDb and is widely regarded as one of the greatest movies of all time, so it's definitely worth checking out.

Generic Explanation

The Shawshank Redemption is a must-watch movie for anyone who loves a good story. It's a powerful and emotional tale of hope, friendship, and redemption that will leave **you** feeling inspired. The film is beautifully shot and features outstanding performances from its cast, including Tim Robbins and Morgan Freeman. The story is gripping and will keep **you** on the edge of your seat until the very end. If **you're** looking for a movie that will make **you** think, feel, and appreciate the power of the human spirit, then The Shawshank Redemption is definitely worth watching.

GPT Disrecommendation

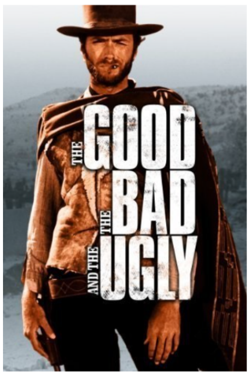
Personalized Explanation

Someone with a preference for action movies like The Fast and the Furious, John Wick: Chapter 3 - Parabellum, and Star Wars: Episode I - The Phantom Menace, and a dislike for movies like Spider-Man: Homecoming, Batman Begins, and Saving Private Ryan, may not enjoy Twilight. Twilight is a romantic fantasy movie that focuses on the relationship between a human and a vampire, with a slower pace and less action than the movies mentioned above. Additionally, the themes and tone of Twilight may not align with the preferences of someone who dislikes movies like Spider-Man: Homecoming and Batman Begins. Therefore, it may not be the best choice for someone with these movie preferences.

Generic Explanation

Twilight is a movie that may not be suitable for everyone. The movie is known for its cheesy dialogue, lack of character development, and weak plot. The romance between the main characters is often criticized for being unrealistic and unhealthy. Additionally, the movie has been accused of promoting toxic relationships and perpetuating harmful gender stereotypes. If **you** are looking for a well-written and thought-provoking movie, Twilight may not be the best choice for **you**.

Random Recommendation



Personalized Explanation

The Good, the Bad and the Ugly is a classic western movie that features a thrilling story, iconic characters, and an unforgettable soundtrack. If **you** enjoy movies with a strong sense of adventure and a well-crafted plot, then this movie is definitely worth watching. Additionally, if **you** appreciate movies that have a unique style and a distinct visual flair, then **you** will likely enjoy The Good, the Bad and the Ugly. While this movie may not have any direct similarities to the movies **you** listed as your favorites or dislikes, it is a timeless classic that has stood the test of time and is widely regarded as one of the greatest movies ever made.

Generic Explanation

The Good, the Bad and the Ugly is a classic western movie that is worth watching for its iconic characters, memorable soundtrack, and epic showdowns. The film follows three gunslingers who are in search of a buried treasure during the American Civil War. The movie is known for its stunning cinematography and intense action scenes. If **you** enjoy westerns or are a fan of director Sergio Leone's work, **you** will definitely appreciate this masterpiece. Additionally, the film has influenced many other movies and pop culture references, making it a must-see for any film buff.

Colors descriptions

Argument from comparison
Argument from criterion
Argument *ad populum*
Argument from evaluation

Figure 4.9: User-based vs. Generic Explanations for three recommendations obtained from real users.

Arguments from Evaluation predominantly constitute nearly every statement in the explanations. The examples also indicate the use of *Argumentum ad populum* in two personalized explanations, where the appeal references the popular notion that movies are widely recognized as **some of the greatest films of all time** and have high ratings on sites such as IMDb. *Arguments from comparison* are also present twice in the first example, with the initial comparison emphasizing the positive preferences shared by the user and the specific recommendation. In contrast, the second focuses on the contrasts between the recommendation and the user's negative preferences. Lastly, the example showcases an *Argument from Criterion* in a generic explanation, highlighting that the movie may not be appropriate for the user if there are particular concerns regarding its plot (the criteria).

We also found that both user-based and generic explanations exhibit a persuasive tone, primarily through the frequent use of second-person language. In Figure 4.9, we emphasized the pronoun **you** in bold to highlight its common usage. The frequent appearance of this phrasing could stem from the prompt used for generating the explanation, which encourages the model to address the audience directly, or from the model's training to engage in this communication style when interacting with people. This finding might clarify why we did not find notable differences in the persuasiveness between the two explanation types (refer to Section 4.2.2).

We also examined the co-occurrence patterns of arguments within the explanations. Initially, we focused on analyzing the frequency of argument trigrams. We chose to analyze trigrams as three sentences were the median size of the explanations. Our findings indicated that in generic explanations, combinations of sentences without arguments and those utilizing *Arguments from Evaluation* appear in more than 70% of instances, which aligns with the high occurrence of these sentence types, as shown in Figure 4.8. In contrast, user-based explanations displayed a more uniform distribution, with the most common combination occurring in less than 5% of cases. Looking at the top-10 argument chains for user-based explanations, they consist of a mix of sentences lacking arguments or employing *Arguments from Similarity*, *Arguments from Comparison*, *Arguments from Evaluation*, and *Arguments from Opposites*. Figure 4.10 illustrates the Top-10 most frequent trigrams for both explanation types.

The analysis of trigrams indicates that generic explanations can be persuasive due to a

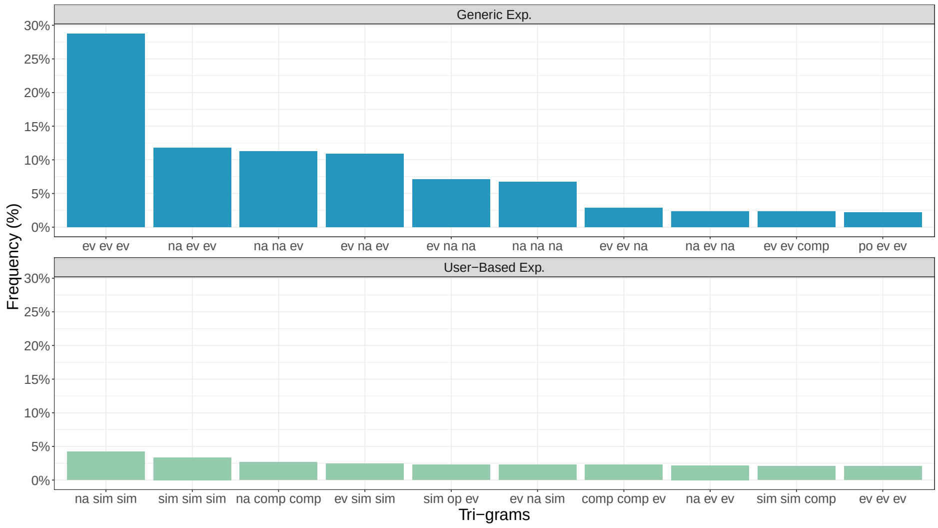
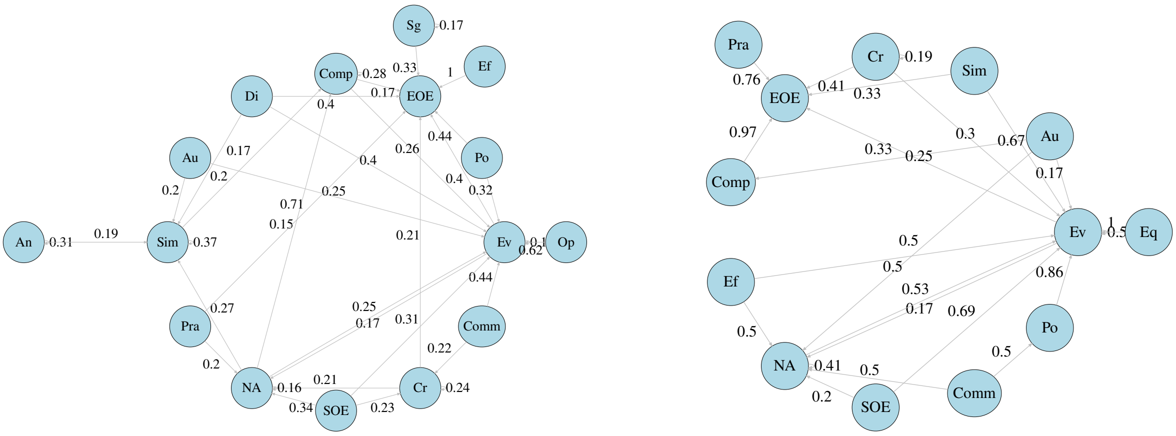


Figure 4.10: Frequency of the top-10 trigrams of types of arguments found in the explanations grouped by the type of explanation. Classifications codes: na (No Argument), comp (Argument from Comparison), ev (Argument from Evaluation), op (Argument from Opposites), sim (Argument from Similarity), po (Argumentum ad Populum).

blend of universally understandable, consistent, and evident argument formats. They create a refined and coherent presentation that resonates with a broad audience by highlighting straightforward patterns, such as the regular use of evaluative statements. This simplicity ensures that the content is accessible while maintaining a level of sophistication that enhances its persuasive and relatable qualities. Additionally, the frequent use of second-person language provides a sense of personalization, even without direct customization. These aspects may account for the perception that generic explanations are just as effective, persuasive, and personalized as those explicitly tailored to users, especially in situations where universal appeal and clarity are adequate to fulfill user expectations.

Finally, we investigated the transition between arguments in each type of explanation (generic and user-based). Using a first-order Markov Chain, we created transition probability graphs, shown in Figure 4.11. To enhance readability, we filtered the edges for transitions with a strength of 0.15 or greater. Our findings highlight notable differences in how arguments are organized within the two explanation types. In user-specific explanations, the transitions are varied and do not favor any single pattern, suggesting a more complex structure that is tailored to the individual context of each user. This intricate organization supports more nuanced reasoning, making these explanations particularly engaging, especially when presenting novel or unexpected recommendations.



(a) User-Specific Explanations

(b) Generic Explanations

Figure 4.11: Comparison of Markov Chains for User-Specific and Generic Explanations. Transition edges were filtered to transitions equal or higher than 0.15 to improve readability. The nodes SOE and EOE were included to indicate, respectively, the start and end of the explanations.

In comparison, generic explanations follow more straightforward, linear transitions, frequently alternating between evaluative arguments and factual statements (*No Argument - NA*). Similarly to what we found in the trigram analysis, this consistent structure makes it easily accessible to a broad audience. Furthermore, universal appeals—like practical arguments and those referencing popularity—boost their relatability and perceived effectiveness, even without personalization elements.

In general, user-specific explanations are more engaging due to their dynamic nature, making them particularly effective when dealing with unfamiliar or random recommendations. On the other hand, generic explanations stand out for their simplicity and consistency, making them relatable and bridging the perceived effectiveness gap between the two types. These differences clarify **RQ5** and explain why users may find both types of explanations effective, with their preferences varying based on the context and the type of recommendations provided.

4.2.7 Popularity Bias in GPT-generated Recommendations

Although evaluating the quality of the GPT model's recommendations was not our primary focus, we observed a notable influence of popularity bias. This bias affects movies with exceptionally high or low IMDb ratings, leading to their over-representation in recommendations and disrecommendations.

We found evidence supporting this observation. A positive, though weak to moderate, correlation exists between the frequency of recommendations and IMDb ratings in GPT-generated recommendations (Spearman's $\rho = 0.467$, $p < 3 \times 10^{-5}$; Kendall's $\tau = 0.383$, $p < 3\times10^{-5}$). In contrast, a weak negative correlation was observed for disrecommendations (Spearman's $\rho = -0.268$, $p < 0.05$; Kendall's $\tau = -0.205$, $p < 0.05$). No significant correlation was detected for random recommendations.

These findings suggest that the recommendations generated by the GPT model may be influenced by popularity bias, but further analysis is needed to explore this bias's implications on the recommendations' overall performance.

4.3 Overview

In this chapter, we investigated the capabilities of LLMs in effectively generating automated human-centered explanations for Recommender Systems by analyzing the perception of users concerning the satisfaction with the recommendation and the effectiveness, personalization, and persuasiveness of the provided explanation. We analyzed this in three dimensions: familiarity, personalization, and intention of the recommendation (positive or negative recommendation). This analysis contributed by expanding the related literature based on LLM-Recs that only focused on evaluating the accuracy of recommendations through offline experiments such as Gao et al. [13] and Harrison, Dereventsov, and Bibin [14].

Thus, we can summarize our findings in the following:

1. Personalized recommendations from ChatGPT yielded higher user satisfaction than random (but popular) recommendations;

2. Personalized explanations were not perceived as significantly more personalized than generic explanations unless the recommendations were randomly generated,

even with them referring to the users' preferences explicitly;

3. User-based explanations were observed as, in some measure, more effective, personalized, and persuasive for unfamiliar movies, perhaps because prior knowledge about movies had less influence on decision-making, leaving more room for explanations to influence users' choices;

4. Regarding disrecommendations, explanations did not benefit from user-based explanations.

5. Our path modeling analysis revealed the interdependence between the different recommendation goals, with the explanation effectiveness being strongly predicted by users' satisfaction, persuasiveness, and personalization perceptions, with persuasiveness exerting the most significant influence.

6. Our analysis showed evidence of a *popularity bias* in the recommendations made by the GPT model, although a broader study is still necessary to verify this.

7. We shed light on interesting differences in how the model constructed the generic and user-based explanations in terms of argumentation, revealing that the simplicity of the argumentation in the generic explanations helped to bridge the gap in perceived effectiveness, even in scenarios where explicit personalization was expected to dominate.

The experiment and results discussed in this chapter were published in a paper entitled *Leveraging ChatGPT for Automated Human-centered Explanations in Recommender Systems* [36] presented at the 29th International Conference on Intelligent User Interfaces (IUI 2024) held in Greenville, SC, USA in March 2024. An extended version of this paper was submitted to ACM Transactions on Interactive Intelligent Systems (ACM TiiS) in a special issue for "IUI 2024 highlights" and is under review.

4.3.1 Limitations

While this analysis demonstrated significant effects of GPT-generated explanations, it has some limitations. Firstly, we used only one LLM, specifically the GPT 3.5 Turbo, which was motivated by its ease of use provided by its API, which required minimal hardware compared to other LLMs that, by the time, were not available through APIs, such as Meta's Llama and Google's Gemma. This narrowed focus limits the generalization of our findings.

Other limitations stem from the nature of LLMs and are expected and inherent to this type of technology. When interacting with these models, the formulation of the prompt can significantly affect the outcome. To minimize the impact of variations in output due to differences in prompt formulation, we created a standardized set of prompts (see Figure 4.2) for our study.

Another limitation of our analysis is the arguably limited room for personalization, which was restricted to six preferences (three liked and three disliked movies). While for a traditional recommendation model, a modest number of preferences suffice to generate reasonable recommendations [31], we have no insight into the finer details of how our ChatGPT-based recommender creates recommendations. However, it should be noted that this experiment focused on comparing personalized (user-based) and generic explanations. With that in mind, even a relatively modest level of personalization should suffice.

In line with the limited insight into the delicate workings of ChatGPT's recommendation process, another limitation arises from the distinction between explanation and justification. In this context, an explanation can be understood as revealing the algorithmic reason behind a recommendation, while a justification refers to motivating why an item is recommended [11]. Considering the recommendation model used in our study, the explanations generated by our system are rather justifications, i.e., human interpretable snippets informing the study participants why a particular item was recommended, than explanations. However, in the context of this analysis, the semantic differences between the terms explanation and justification should not affect the outcomes. We further explore whether the explanations provided by LLM-Recs are faithful (i.e., indeed describe the model internal reasoning) in our **RQ2** and Chapter 5.

Lastly, regarding the classification of arguments, we cannot guarantee that GPT-4o is a good classifier due to the lack of a labeled dataset containing all the classes that could be used for evaluation. However, it is well known for good classification performance in several domains. A future study could use the Argument Type Identification Procedure (ATIP) [43] to increase accuracy and confidence through a step-by-step procedure.

Chapter 5

Faithfulness in Black-Box LLM-based Recommender Systems' Explanations

In this chapter, we outline the experiment performed to evaluate the faithfulness of explanations generated by LLM-based Explainable Recommender Systems (LLM-RS), aligned with our **RQ6** and to address the limitations presented in our previous experiment. For this analysis, we investigated the explanations provided by four LLM-Rec in three different domains, considering recommendations and disrecommendations. The first section details the experimental design (metrics, models, and datasets). The second section presents and discusses the results. Finally, the last section contains an overview of the results and their contributions and limitations.

5.1 Experiment Design

5.1.1 Mathematical Formulation

Formally, let U be a set of users and I a set of items. A recommender system $\mathcal{R}$ can be interpreted as a function that, given a user $u \in U$, outputs a set of recommended items $R_u \subseteq I$. For this experiment, the role of the recommender system was performed by an LLM, denoted as $\mathcal{L}$. The LLM received, via prompt, a set of information or preferences (I_u) about a given user ($u \in U$) and was tasked with recommending items that the user may or may not be interested in. A recommendation is considered *positive* if it is intended to generate interest in the user, and *negative*[1] otherwise. As this experiment did not require an online evaluation, we expanded the scope to include three domains—movies, songs, and books—and four models: GPT4o, Gemma 2 9B, Llama 3 70B, and Mixtral 8x7B. The set of items consisted of all items within these domains that the LLMs learned during the pre-training.

For each recommendation, the LLM also provided a self-generated explanation of why that recommendation was made. To evaluate the faithfulness of these self-explanations, we adopted an *axiomatic approach* by adapting the *Feature Importance Agreement* [48], which states that if the predicted label of a classification and its rationale are associated, then the input tokens that are important for the label prediction should also be important for the rationale generation. In our context, we rephrase this as the following: *If a user preference is important for the recommended item, then it should be present in the explanation with equal importance*. The remainder of this section discusses how to obtain the input items[2] importance, concerning both recommendation and explanation and which metrics we used to compare them.

5.1.1.1 Obtaining user's preferences importance for the recommendation

Since we treated the LLMs as opaque models, the only elements we can manipulate are the inputs of the model, and the only elements we can observe are its outputs, which required our method to be model-agnostic. In the interpretable machine learning literature, two model-agnostic methods stand out for input feature scoring: LIME [32] and SHAP [23].

LIME works by fitting an explainable surrogate model to perturbations of the input and the corresponding predictions from the black-box model, then locally explaining the input based on the surrogate model's interpretation. On the other hand, SHAP applies the concept of Shapley Values from game theory to determine the contribution of each input feature to a given prediction. For this study, we developed our solution based on Shapley Values due to their desirable properties (detailed in Section 2.4) and the difficulty in fitting an explainable surrogate model for recommendations, which does not allow us to use LIME. However, differently from SHAP and CC-SHAP [29], we will not use an approximation of Shapley

[1] In this work, we also used the term disrecommendation to indicate negative recommendations.

[2] In the remainder of this chapter, we will use user preference and input item as equivalent terminologies.

Values but instead calculate it considering all the possible coalitions, i.e., subsets of features, which guarantees their values to be *faithful*.

Computing Shapley Values In our recommendation context, the set of players N is equivalent to the input set passed to the recommender. Therefore, the coalitions are subsets of the input set. Considering $R_u^{\mathcal{L}}$ as the recommendations made by the recommender $\mathcal{L}$ for user u and the input set of items I_u, for each $r \in R_u^{\mathcal{L}}$, we define G_r as the coalitional game $G_r(I_u, f_r)$, where:

$$f_r(S) : 2^{I_u} \to \{0,1\} = \begin{cases} 1 & r \in \mathcal{L}(S) \\ 0 & \text{otherwise} \end{cases}$$

Thus, to determine the importance of an item ($i \in I_u$) with respect to the recommendation r for user u, we compute its Shapley Value for game $G_r(I_u, f_r)$ by adapting Equation (2.1):

$$\varphi_i^r = \frac{1}{n}\left(\sum_{S \subseteq I_u \setminus \{i\}} \frac{1}{C_{n-1}^{|S|}} [f_r(S \cup \{i\}) - f_r(S)]\right) \tag{5.1}$$

5.1.1.2 Obtaining user's preferences importance for the explanation

Since our explanations are textual, we computed two textual-based importance scores: *Citation* and *LLM-ranking*.

Citation This metric is based on the premise that if an item is cited in the explanation, then it is important. Thus, let E_r represent the natural language explanation provided for recommendation r, for each input item $i \in I_u$, we compute the *Citation* of i with respect to E_r as

$$c(i, E_r) = \begin{cases} 1 & i \in E_r \\ 0 & \text{otherwise} \end{cases}$$

For example, consider that a user u has interacted with a set of input items $I_u = \{\text{item}_1, \text{item}_2, \text{item}_3\}$, and the system recommends $r = \text{item}_4$. The system provides the following natural language explanation for the recommendation: *We recommend item*$_4$ *because it is similar to item*$_1$ *and item*$_3$*, which you have previously engaged with.* To compute the Citation metric, we check if each input item $i \in I_u$ is explicitly mentioned in the explanation

E_r. For item$_1$, since it is mentioned in E_r, $c(\text{item}_1, E_r) = 1$. Similarly, item$_3$ is also mentioned, so $c(\text{item}_3, E_r) = 1$. However, item$_2$ is not mentioned, resulting in $c(\text{item}_2, E_r) = 0$. Therefore, the Citation values for the input items are $c(\text{item}_1, E_r) = 1$, $c(\text{item}_2, E_r) = 0$, and $c(\text{item}_3, E_r) = 1$.

LLM-ranking Based on the idea that an item may be mentioned implicitly, which would not be covered by the previous metric, we prompted GPT-4o with the following task: *given a set of previously consumed items by a user, and the explanation for a recommended item, rank these items according to the importance given to them by the explanation* (Appendix A.3). Thus, for E_r, we obtain R_{E_r}, which is the ranking for each $i \in I_u$.

5.1.1.3 Faithfulness evaluation

To evaluate the faithfulness of the models' explanations, we compared the item importance obtained for the recommendation with the one obtained from the explanation. In this section, we define the metrics we used for that.

Recall & Precision Let Φ_r^+ be the set of items with the Shapley Values larger than 0.1 for recommendation r and C^r be the set of items cited in E_r, we define:

$$Recall = \frac{|C^r \cap \Phi_r^+|}{|\Phi_r^+|} \text{ and } Precision = \frac{|C^r \cap \Phi_r^+|}{|C^r|}$$

The *Recall* can be interpreted as the percentage of important items for the recommendation cited in the explanation. Meanwhile, the *Precision* can be interpreted as the proportion of cited items important for the recommendation. For example, consider a scenario where a recommendation r was made and the explanation E_r cited a set of items $C^r = \{\text{item}_1, \text{item}_3\}$. Additionally, $\Phi_r^+ = \{\text{item}_1, \text{item}_2, \text{item}_3\}$, which includes items with Shapley Values larger than 0.1. To compute Recall and Precision, we first determine the intersection of C^r and Φ_r^+, which is $C^r \cap \Phi_r^+ = \{\text{item}_1, \text{item}_3\}$. Then, the *Recall* is calculated as $\frac{|C^r \cap \Phi_r^+|}{|\Phi_r^+|} = \frac{2}{3}$, and the *Precision* is $\frac{|C^r \cap \Phi_r^+|}{|C^r|} = \frac{2}{2} = 1$.

Weighted Coverage Let $R_{E_r}^k$ be the top-k ranked item from the *LLM-ranking* for recommendation r, we define:

$$WCov_r@K = \sum_{n=1}^{K} \frac{1}{n} \phi^r_{R^n_{E_r}}$$

This can be interpreted as the amount of importance covered by the top-K items ranked according to the recommendation's explanation. The importance is weighted by position so that if the item with a higher Shapley Value appears in a low-ranking position, its contribution to the metric is attenuated.

Given the scenario where a recommendation r was made, and the *LLM-ranking* produced the top-3 ranked items $R^k_{E_r} = [\text{item}_1, \text{item}_3, \text{item}_2]$. The Shapley Values for these items with respect to r are $\phi^r_{\text{item}_1} = 0.4$, $\phi^r_{\text{item}_3} = 0.3$, and $\phi^r_{\text{item}_2} = 0.2$. To compute the Weighted Coverage for $K = 3$, we use the formula:

$$WCov_r@3 = \sum_{n=1}^{3} \frac{1}{n} \phi^r_{R^n_{E_r}} = \frac{1}{1}\phi^r_{\text{item}_1} + \frac{1}{2}\phi^r_{\text{item}_3} + \frac{1}{3}\phi^r_{\text{item}_2}$$

Substituting the values:

$$WCov_r@3 = (1)(0.4) + \left(\frac{1}{2}\right)(0.3) + \left(\frac{1}{3}\right)(0.2) = 0.4 + 0.15 + 0.0667 = 0.6167$$

Thus, the Weighted Coverage $WCov_r@3$ is 0.6167, reflecting the importance covered by the top-3 ranked items, where higher-ranked items and those with greater Shapley Values contribute more to the metric.

5.1.2 The datasets

We selected three distinct domains — movies, music, and books. These domains are recurring in the recommendation systems literature and will enhance the robustness of our results and evaluate the models' faithfulness across diverse contexts. For each domain, we chose a dataset. In the following, we provide a detailed description of each dataset and the corresponding preprocessing steps.

IUI 2024 Movie Preferences This is the dataset from the experiment described in Chapter 4 [36] and contains the movie preferences of 94 users obtained through the survey. Each user's preferences consist of three liked and three disliked movies.

Spotify Million Playlist Dataset The Spotify Million Playlist Dataset (SMPD) [8] was released in 2018 as part of the ACM Recommender Systems Challenge 2018 and is now available on Kaggle[3]. It contains data on one million playlists sampled from Spotify[4], including playlist names, the number of tracks, the number of distinct albums and artists, and the tracks themselves. For our experiment, we randomly sampled 100 playlists from the dataset, extracting the title of each playlist and the title and artist of its first five tracks. Playlists with fewer than five tracks were excluded prior to sampling.

Book Recommendation Dataset The Book Recommendation Dataset is available on Kaggle[5] and was obtained from the Book-Crossing community[6]. It contains data from over 270,000 users, with more than one million ratings for approximately 270,000 books. The ratings range from 0 to 10, where 0 indicates a book the user did not evaluate, and the other values represent the user's rating for the book. We excluded books with a zero rating for this experiment since they represent implicit feedback and do not indicate a user's disinterest. We randomly selected 100 users from the dataset and randomly sampled three liked and disliked books for each. A book was considered liked by a user if it had a rating above the median rating of the user; otherwise, it was considered disliked.

For both the SMPD and Book Recommendation datasets, we selected 100 samples to maintain consistency with the number of users in the IUI 2024 Movie Preferences dataset and because conducting the experiment with a larger sample size would involve significant time and financial costs.

5.1.3 The models

For this experiment, we followed the paradigm LLMs as Recommender Systems. In this paradigm, a pre-trained LLM generates recommendations by processing user demographic and/or consumption data [51]. To ensure diversity and enable comparisons between closed and open-source models, we selected four of the most popular LLMs available: the closed-

[3]https://www.kaggle.com/datasets/himanshuwagh/spotify-million
[4]https://www.spotify.com
[5]https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset
[6]https://www.bookcrossing.com

source *OpenAI GPT-4o*[7] and the three open-sourced models, *Gemma 2 9B*[8], *Meta Llama 3 70B*[9] and *Mixtral 8x7B*[10]. The OpenAI's GPT-4o is an improvement on the GPT-3.5 Turbo used in the experiment of Chapter 4.

All four models were accessed via APIs using Python libraries: *openai-python*[11] for the GPT-4o and *groq-python*[12] for the others. Since these models are accessed through APIs, they were treated as opaque in this study. Although the open-source models are available for download and local deployment, we accessed them through APIs to avoid the high infrastructure costs associated with local runs.

5.1.4 Recommendation tasks

Each model was assessed on three recommendation tasks, corresponding to the three datasets, with unique prompts for each task. Each task had two prompts: one for obtaining the recommendations and the other to obtain the explanation of a given recommendation. We chose to use two prompts to reduce the costs since the recommendation prompt was also used to get the recommendations for each coalition of inputs for which we did not need the explanations[13]. The same prompts were used uniformly across all models for a given task. Each model was asked to recommend items for each user of each dataset. For each recommendation, an explanation was also requested. We describe the tasks in Table 5.1.

5.2 Results & Discussion

5.2.1 Recall & Precision

Table 5.2 summarizes the performance of the four models — GPT-4o, Gemma2 9B, Llama3 70B, and Mixtral 8x7B — on the three datasets (Book, Our, and SMPD) using average

[7]https://openai.com/index/hello-gpt-4o/

[8]https://huggingface.co/google/gemma-2-9b-it

[9]https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct

[10]https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

[11]https://github.com/openai/openai-python

[12]https://github.com/groq/groq-python

[13]Considering that in the experiment of the previous chapter, we did not see any difference between prompt for the recommendation and explanation together and separated, this could be done without harming the experiment validity.

Table 5.1: Recommendations tasks, their input, and prompts.

Task name	Task description	Inputs	Prompts
Movie recommendation	The model is asked to recommend two movies the user may like and two they may dislike, along with an explanation	Three liked and three disliked movies	Prompts A.2.1 and A.2.2
Playlist completion	The model is asked to recommend two songs the user may include and two they may not include in their playlist, along with an explanation	The title of the playlist and the first five tracks (name and artist)	Prompts A.2.3 and A.2.4
Book recommendation	The model is asked to recommend two books the user may like and two they may dislike, along with an explanation	Three liked and three disliked books	Prompts A.2.5 and A.2.6

precision (AP), average recall (AR), and the F1-score, computed over the AP and AR, for both positive and negative recommendations.

The results reveal differences across all three study dimensions: type of recommendation (positive or negative), domain, and models. Considering the type of recommendation, negative recommendations consistently showed the lowest average performance across all metrics, models, and domains. Similarly to what we found about disrecommendation in our previous analysis (Chapter 4), this may be due to the reluctance of these models to produce negative discourse, which, in turn, may lead to the disliked movies not being cited in the explanations.

As for the varying performance across datasets, it suggests that some domains are more susceptible to *unfaithful* explanations. The Book dataset performed worst among the datasets, with low F1-scores across all models. In contrast, the IUI24 dataset achieved the highest scores, indicating that the model has better domain knowledge. The SMPD dataset demonstrated intermediate performance, with models attaining moderate F1-scores. A plausible reason for this may be that the amount of inherent knowledge (data) of the models about certain domains may differ. However, a specific experiment would be necessary to confirm this.

Regarding the differences between the models, we can see that GPT-4o demonstrates superior F1-scores for both positive and negative recommendations across all datasets. Mixtral 8x7B, while slightly behind GPT-4o, consistently ranks second. Llama3 70B and Gemma2 9B sometimes show competitive performance, but their overall scores remain lower than those of GPT-4o and Mixtral 8x7B. These results suggest that some models perform better in terms of faithfulness than others.

Additionally, we also observed higher values of AP across all the domains and models in comparison with AR, thus suggesting that the explanations cite important items with a certain frequency, although they do not cover all the important items.

One drawback about this metric is that it does not consider the importance rank of the items. Thus, a model can achieve good results without, for example, citing the most important item. Also, this metric ignores implicit citations of the input items since it is based on textual occurrence (see Table 5.3). Thus, it leads to the need for a more contextualized metric such as Weighted Coverage.

5.2.2 Weighted Coverage

As detailed previously, the Weighted Coverage is computed considering the rank obtained by GPT-4o from the explanation and the computed Shapley Values of the input items. Figure 5.1 illustrates the distribution of Weighted Coverage at Top-3 (WCov@3) for both positive

Table 5.2: Average Precision and Recall

		Positive Recs.			Negative Recs.		
Model	**Dataset**	**AR**	**AP**	**F1**	**AR**	**AP**	**F1**
GPT-4o	Book	0.192	0.275	0.226	0.380	0.569	0.456
	IUI24	0.769	0.721	0.744	0.661	0.698	0.679
	SMPD	0.660	0.662	0.661	0.571	0.664	0.614
Gemma2 9B	Book	0.132	0.238	0.170	0.282	0.455	0.349
	IUI24	0.692	0.663	0.677	0.434	0.475	0.453
	SMPD	0.393	0.577	0.468	0.147	0.313	0.200
Llama3 70B	Book	0.198	0.284	0.233	0.298	0.507	0.376
	IUI24	0.702	0.721	0.712	0.487	0.724	0.582
	SMPD	0.639	0.625	0.632	0.287	0.507	0.366
Mixtral 8x7B	Book	0.277	0.494	0.355	0.309	0.682	0.426
	IUI24	0.672	0.613	0.641	0.559	0.653	0.602
	SMPD	0.559	0.804	0.659	0.431	0.774	0.554

Note:

AR = Average Recall

AP = Average Precision

F1 = F1-score of Average Recall and Precision

(TRUE) and negative (FALSE) recommendations across the four models — Gemma2 9B, GPT-4o, Llama3 70B, and Mixtral 8x7B — and the three datasets: Book, IU24, and SMPD. The red dashed line represents a reference value (0.5) established based on observations made during the analysis of the data. Items' explanations with a score lower than 0.5 were observed to be highly *unfaithful*. Meanwhile, the explanations with a score above seemed more *faithful*.

Due to the novelty of the method applied, especially in the recommendation scenario, we could not find a comparable method to use as our comparison baseline. Other model-agnostic faithfulness tests, such as input perturbation via counterfactual and adversarial examples, are not easy to apply since obtaining a counterfactual for a movie, for example, is not straightforward and would require a helper model trained on task-specific data. In turn, input reconstruction methods would be easy for the model to answer since the input items are usually cited in the explanation.

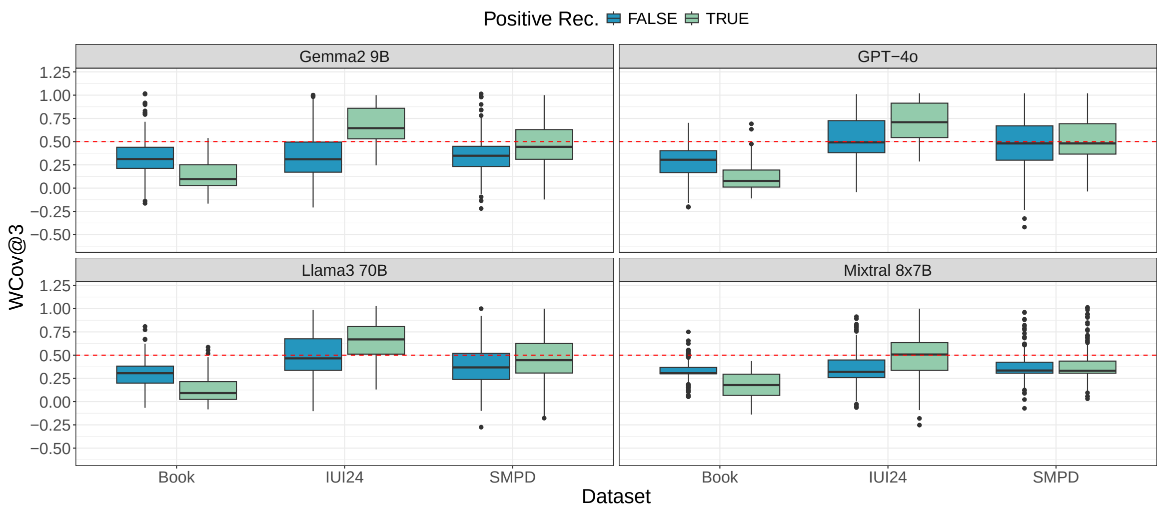


Figure 5.1: Distribution of the Weighted Coverage at Top-3 for the four models and three datasets

Across all models and datasets, there is a notable distinction between the distributions for positive and negative recommendations, with the positive generally achieving higher WCov@3 values, except in the case of the Book dataset, where the opposite occurs, which indicates that the models are more effective in producing faithful explanations for positive recommendations. Again, this may be due to the models' reluctance to produce negative comments on some topics, which may lead them to not using the negative preferences when explaining a negative recommendation.

The IUI24 dataset yields the highest median WCov@3 values across most models. Conversely, the Book dataset shows lower WCov@3 values for recommendations and disrecommendation, with all the models having at least 75% of the explanations below the reference value. The SMPD dataset shows moderate performance, with broader variability in WCov@3 values across all models. This aligns with the results for Precision and Recall and indicates the relation between the model's knowledge of a specific domain and its capabilities to provide faithful explanations.

Table 5.3: Example of Mixtral 8x7B's Movie Recommendation

Model	**Positive Rec.**	**Recall**	**Precision**	**WCov@3**
Mixtral 8x7B	Yes	0.75	0.6	-0.181

Explanation	
	The Godfather is a classic crime film known for its strong characters and intricate plot, which you may enjoy based on your preference for action-packed movies like Terminator 2: Judgment Day and Charlie's Angels. It is not a horror movie, which aligns with your dislike of Scary Movie, Scream, and Friday the 13th.

GPT Ranking	**Liked**	**Name**	**Shapley Value**
1	Yes	Terminator 2: Judgment Day	-0.250
2	Yes	Charlie's Angels	-0.150
3	No	Scary Movie	0.433
4	No	Scream	0.383
5	No	Friday the 13th	0.133
6	Yes	Thelma & Louise	0.450

Regarding model-specific observations, GPT-4o consistently exhibits higher median WCov@3 values for positive recommendations across all datasets, particularly in the IU24 dataset, surpassing other models. Llama3 70B and Gemma2 9B perform similarly in all the datasets; both are slightly better than Mixtral 8x7B, which displays lower medians overall.

This contrasts with the previous metrics, where Mixtral 8x7B was the second best, which indicates that although relevant items are used in the explanations, their relevance in the explanations may not align with the input-item relevance.

Table 5.3 shows an example of Mixtral 8x7B's movie recommendation with good scores in the Recall and Precision but with a negative score for the WCov@3. The explanation uses the liked movies to justify the recommendation, which is expected since it is a positive recommendation. However, looking at the Shapley Values, the explanation cites two liked movies with negative scores, indicating that they did not contribute to the recommendation. Moreover, it leaves out the positive movie with the highest score, indicating an *unfaithful* explanation. In contrast, Table 5.4 shows a book disrecommendation example from the Llama model, in which both the perceive importance extract from the explanation and the Shapley Values align. The explanation starts by highlighting the differences of the disrecommended book with the books the user liked, which include the one with the highest Shapley Value.

5.3 Overview

In this chapter, our goal was to address **RQ6**, which intended to verify if the explanations given by LLM-based Explainable Recommender Systems are faithful. In order to do this and taking into account that we were using these models as black boxes, we opted for an *axiomatic test*. Considering the *Feature Importance Agreement* [48], we investigated whether the importance of the input items for the recommendation and its explanation were aligned.

We used Shapley Values to compute the input items' importance for the recommendation and considered two ways of obtaining the input importance regarding the explanation: citation and LLM-based ranking.

The results highlighted differences between models, domains, and the type of recommendation (positive or negative). Regarding the models, GPT-4o consistently achieved better metrics across all domains. Domains' differences were perceived in the four models, with the movie domain leading to the better metrics and the book domain to the worst, which, as discussed, may be due to the knowledge of the models. Explanations for positive recommendations also showed higher scores overall, which may be caused by the models' intrinsic instructions to avoid damaging or harmful speeches.

Table 5.4: Example of Llama3 70B's Book Recommendation

Model	Positive Rec.	Recall	Precision	WCov@3
Llama3 70B	No	1	0.25	0.772

Explanation	Based on the user's reading preferences, it is likely that they will not enjoy 'The Devil Wears Prada' by Lauren Weisberger. The user has shown a strong interest in non-fiction books that focus on science, technology, and engineering, as seen in their liked books 'The CASE FOR MARS', 'Modern Operating Systems', and 'Applied Cryptography'. In contrast, 'The Devil Wears Prada' is a work of fiction that focuses on fashion and relationships, which is a significant departure from the user's preferred genres. Additionally, the user has disliked books that are more focused on personal stories and relationships, such as 'What Not to Wear' and 'Raj: A Novel', which further suggests that they may not enjoy 'The Devil Wears Prada'.

GPT Ranking	Liked	Name	Shapley Value
1	Yes	The CASE FOR MARS - by Robert Zubrin	0.717
2	Yes	Modern Operating Systems - by Andrew S. Tanenbaum	0.100
3	Yes	Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition - by Bruce Schneier	0.017
4	No	What Not to Wear - by Susannah Constantine	0.100
5	No	Raj: A Novel - by Gita Mehta	0.050
6	No	Man Who Mistook His Wife for a Hat - by Oliver Sacks	0.017

Overall, the results indicated that LLMs do not consistently generate faithful explanations for recommendations, as the explanations did not satisfy the axiomatic test in all the cases, which answers **RQ6**. These findings indicate that LLM-generated explanations should be carefully considered since they present a persuasive tone and very plausible reasoning but may mask the actual reasoning.

5.3.1 Limitations

Similarly to the previous analysis, one of this study's limitations comes from the nature of the non-deterministic nature of the LLMs. Different prompt formulations could significantly affect the results. To mitigate the impacts of this characteristic, we consistently used the same prompt for a given task across all models. Also, we set the models' temperature to zero to obtain more stable responses.

Another limitation comes from the limited number of input items, which may not provide the model with sufficient information for accurately personalized recommendations. However, given the scope of this analysis, even a relatively modest level of personalization should be enough for the model to generate recommendations and explanations.

Using LLMs as black boxes also brings some limitations, as it limits our access to the models at their input and output, which limits our space to produce metrics and evaluation methods with more insight into the inner processing steps of the models, like their embeddings and output token probabilities. However, we chose to deal with them as black boxes to produce a model-agnostic technique since one of the current most powerful models (GPT-4o) is a closed-source model.

As stated previously, there's no sufficient test for attesting the faithfulness of natural language explanations, so we could not fully answer our research question. However, we were able to provide some necessary conditions for a faithful explanation in the recommendation scenario through the Weighted Coverage metric, with an explanation needing to achieve at least 0.5 in the WCov@3 to not be considered unfaithful. We also provided insights regarding the capabilities of four popular models in different domains and types of recommendations.

Lastly, *Citation* and *LLM-based Ranking* were used to compute the importance of the input items with respect to the textual explanations. However, these metrics do have their limitations. Since the *Citation* metric was computed using text matching,minor spelling variations of items mentioned in the explanation could have resulted in lower scores. Regarding the *Ranking*, the specific criteria used by the LLM remain unclear, making it difficult to guarantee the accuracy of the results.

Chapter 6

Conclusions

This thesis set out to explore the capabilities of Large Language Models (LLMs) in generating automated, human-centered explanations for recommender systems, as well as their faithfulness in reflecting the internal reasoning of these models. To address this, we elaborated six research questions (**RQ1** to **RQ6**) aiming to shed light on the practical utility, user perception, and limitations of LLM-based Explainable Recommender Systems.

Our findings demonstrate that LLMs, specifically the GPT-3.5 Turbo, can generate recommendations that enhance user satisfaction compared to random (but popular) recommendations. Furthermore, user-based explanations were perceived as somewhat more effective, personalized, and persuasive for unfamiliar movies. Through a path modeling analysis, we detailed the interdependence between the different goals of an explanation - satisfaction, persuasiveness, and personalization - and how these goals can strongly predict the effectiveness of the explanations. We also observed evidence of *popularity bias* in the recommended items, with high and low IMDb ratings favored for positive and negative recommendations, respectively. These findings were compiled in a paper entitled *Leveraging ChatGPT for Automated Human-centered Explanations in Recommender Systems* [36] presented at the 29th International Conference on Intelligent User Interfaces (IUI 2024).

Additionally, we analyzed the types of arguments presented in the explanations. We noticed that most of the arguments were *Arguments from Evaluation*, which justified watching the movies by a statement of value about the movie. Our analysis examined how the model generated both generic and user-specific explanations through argumentation. It demonstrated that the straightforwardness of the argumentation in the generic explanations helped narrow the gap in perceived effectiveness, even in contexts where explicit personalization was anticipated to prevail. These findings were included in an extended version of the paper and submitted to ACM Transactions on Interactive Intelligent Systems (ACM TiiS) in a special issue for "IUI 2024 highlights", which is under review.

Regarding the faithfulness of explanations, we adopted an axiomatic approach using the *Feature Importance Agreement* and found partial alignment between the importance of the input items for recommendations and explanations by comparing Shapley Values for recommendation input importance against citation- and LLM-based rankings for explanations. Key insights include: (1) The results indicated that LLMs do not consistently generate faithful explanations for recommendations, as the explanations frequently did not meet the criteria of the test; (2) GPT-4o emerged as the model with less unfaithful explanations, to the extent of our test, across all the investigated domains, particularly excelling in positive recommendation scenarios; (3) Variability in performance was observed across models, domains, and recommendation types.

Contributions This thesis contributes to the growing body of work on LLM-based recommender systems by providing: (1) A user-centric evaluation framework of explanation quality, expanding beyond offline accuracy metrics; (2) A new dataset regarding LLM-based recommendations and users perceptions; (3) An axiomatic method for assessing explanation faithfulness, adaptable across models and domains; (4) Insights into user preferences, explanation types, and the interplay between different goals of recommendation systems; (5) Details regarding the types of arguments that form the explanations in the scenarios of generic and user-based explanations.

Reproducibility The code and analysis needed to reproduce the experiments detailed in this work are available on GitHub at `https://github.com/issilva5/llm-recommendations-survey` for the experiments of Chapter 4 and at `https://github.com/issilva5/llm-explanations-faithfulness` for Chapter 5.

Limitations Several limitations emerged throughout the study and were detailed in the chapters when relevant. The main limitations were the output variability due to non-deterministic behavior, restricted personalization stemming from a limited input scope, and the black-box nature of models, which limited the scope of some evaluations. Whenever possible, we attempted to mitigate these limitations. For example, we employed strategies such as controlling the temperature of the LLMs to increase their determinism.

Ethics Statement This work examines how to evaluate faithfulness of self-explanations produced by LLM-based Recommender Systems. Consequently, any errors in our approach might result in unfounded confidence or doubt regarding LLMs. While increased skepticism is unlikely to lead to ethical problems, unwarranted confidence can pose significant risks. Thus, the takeaway from this work is that LLM-Recs should not be presumed to offer faithful explanations.

Future Work Building on the findings and limitations, future work should explore: (1) Broader evaluations using multiple LLMs to generalize findings; (2) Enhanced personalization techniques with larger input scopes and diverse datasets; (3) Development of standardized metrics for faithfulness in natural language explanations; (4) Deeper exploration of the distinction between explanation and justification in LLM-based Recommender Systems.

In conclusion, while LLMs show promise in generating human-centered explanations for recommender systems, challenges remain in aligning user perception with objective faithfulness. This work provides a foundational step toward bridging this gap, with implications for designing and evaluating future Explainable LLM-based Recommender Systems.

References

[1] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

[2] Dzmitry Bahdanau. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[3] Krisztian Balog and Filip Radlinski. "Measuring Recommendation Explanation Quality: The Conflicting Goals of Explanations". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '20. Virtual Event, China: Association for Computing Machinery, 2020, pp. 329–338. ISBN: 9781450380164. DOI: 10.1145/3397271.3401032. URL: https://doi.org/10.1145/3397271.3401032.

[4] M. Brauer and J. J. Curtin. "Linear mixed-effects models and the analysis of nonindependent data: A unified framework to analyze categorical and continuous independent variables that vary within-subjects and/or within-items". In: *Psychological Methods* 23.3 (2018), pp. 389–411. DOI: 10.1037/met0000159.

[5] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[6] J. Bruin. *Introduction to Linear Mixed Models @ONLINE*. Feb. 2011. URL: https://stats.oarc.ucla.edu/other/mult-pkg/introduction-to-linear-mixed-models.

[7] Shuo Chang, F. Maxwell Harper, and Loren Gilbert Terveen. "Crowd-Based Personalized Natural Language Explanations for Recommendations". In: RecSys '16. Boston, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 175–

182. ISBN: 9781450340359. DOI: `10.1145/2959100.2959153`. URL: `https://doi.org/10.1145/2959100.2959153`.

[8] Ching-Wei Chen et al. "Recsys challenge 2018: automatic music playlist continuation". In: *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys '18. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, pp. 527–528. ISBN: 9781450359016. DOI: `10.1145/3240323.3240342`. URL: `https://doi.org/10.1145/3240323.3240342`.

[9] Jin Chen et al. "When large language models meet personalization: Perspectives of challenges and opportunities". In: *World Wide Web* 27.4 (2024), p. 42.

[10] Elizabeth Clark et al. "All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 7282–7296. DOI: `10.18653/v1/2021.acl-long.565`. URL: `https://aclanthology.org/2021.acl-long.565`.

[11] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. "Explainable artificial intelligence: A survey". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2018, pp. 0210–0215. DOI: `10.23919/MIPRO.2018.8400040`.

[12] Abhimanyu Dubey et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[13] Yunfan Gao et al. *Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System*. 2023. arXiv: `2303.14524 [cs.IR]`.

[14] Rachel Harrison, Anton Dereventsov, and Anton Bibin. *Zero-Shot Recommendations with Pre-Trained Large Language Models for Multimodal Nudging*. 2023. arXiv: `2309.01026 [cs.AI]`.

[15] S Hochreiter. "Long Short-term Memory". In: *Neural Computation MIT-Press* (1997).

[16] Xu Huang et al. *Recommender AI Agent: Integrating Large Language Models for Interactive Recommendations*. 2023. arXiv: `2308.16505 [cs.IR]`.

[17] Alon Jacovi and Yoav Goldberg. "Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 4198–4205. DOI: 10.18653/v1/2020.acl-main.386. URL: https://aclanthology.org/2020.acl-main.386.

[18] Katikapalli Subramanyam Kalyan. "A survey of GPT-3 family large language models including ChatGPT and GPT-4". In: *Natural Language Processing Journal* 6 (2024), p. 100048.

[19] Bart P. Knijnenburg et al. "Explaining the user experience of recommender systems". In: *User Modeling and User-Adapted Interaction* 22 (4-5 Oct. 2012), pp. 441–504. ISSN: 0924-1868. DOI: 10.1007/s11257-011-9118-4. URL: http://link.springer.com/10.1007/s11257-011-9118-4.

[20] Jianghao Lin et al. *How Can Recommender Systems Benefit from Large Language Models: A Survey*. 2024. arXiv: 2306.05817 [cs.IR].

[21] Junling Liu et al. *Is ChatGPT a Good Recommender? A Preliminary Study*. 2023. arXiv: 2304.10149 [cs.IR].

[22] Hongyu Lu et al. "User Perception of Recommendation Explanation: Are Your Explanations What Users Need?" In: *ACM Trans. Inf. Syst.* 41.2 (Jan. 2023). ISSN: 1046-8188. DOI: 10.1145/3565480. URL: https://doi.org/10.1145/3565480.

[23] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[24] Hanjia Lyu et al. *LLM-Rec: Personalized Recommendation via Prompting Large Language Models*. 2023. arXiv: 2307.15780 [cs.CL].

[25] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. "Towards Faithful Model Explanation in NLP: A Survey". In: *Computational Linguistics* 50.2 (June 2024), pp. 657–723. DOI: 10.1162/coli_a_00511. URL: https://aclanthology.org/2024.cl-2.6/.

[26] David A Magezi. "Linear mixed-effects models for within-participant psychology experiments: an introductory tutorial and free, graphical user interface (LMMgui)". en. In: *Front Psychol* 6 (Jan. 2015), p. 2.

[27] Katie Matton, Robert Ness, and Emre Kiciman. "Walk the Talk? Measuring the Faithfulness of Large Language Model Explanations". In: *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*. 2024. URL: https://openreview.net/forum?id=iqT5Shi1KA.

[28] Tomas Mikolov. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* 3781 (2013).

[29] Letitia Parcalabescu and Anette Frank. "On Measuring Faithfulness or Self-consistency of Natural Language Explanations". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6048–6089. DOI: 10.18653/v1/2024.acl-long.329. URL: https://aclanthology.org/2024.acl-long.329.

[30] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[31] István Pilászy and Domonkos Tikk. "Recommending New Movies: Even a Few Ratings Are More Valuable than Metadata". In: *Proceedings of the Third ACM Conference on Recommender Systems*. RecSys '09. New York, New York, USA: Association for Computing Machinery, 2009, pp. 93–100. ISBN: 9781605584355. DOI: 10.1145/1639714.1639731. URL: https://doi.org/10.1145/1639714.1639731.

[32] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. ISBN: 9781450342322. DOI: `10.1145/2939672.2939778`. URL: `https://doi.org/10.1145/2939672.2939778`.

[33] Howard J Seltman. “Experimental design and analysis”. In: Book is on the World Wide Web, 2018. Chap. 14,15.

[34] Lloyd S. Shapley. “A value for n-person games”. In: *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Ed. by Alvin E.Editor Roth. Cambridge University Press, 1988, pp. 31–40.

[35] Yubo Shu et al. *RAH! RecSys-Assistant-Human: A Human-Central Recommendation Framework with Large Language Models*. 2023. arXiv: `2308.09904 [cs.IR]`.

[36] Ítallo Silva et al. “Leveraging ChatGPT for Automated Human-centered Explanations in Recommender Systems”. In: *Proceedings of the 29th International Conference on Intelligent User Interfaces*. IUI ’24. Greenville, SC, USA: ACM, Mar. 2024, pp. 597–608. ISBN: 9798400705083. DOI: `10.1145/3640543.3645171`. URL: `http://dx.doi.org/10.1145/3640543.3645171`.

[37] Gemma Team et al. “Gemma 2: Improving open language models at a practical size”. In: *arXiv preprint arXiv:2408.00118* (2024).

[38] Gemma Team et al. “Gemma: Open models based on gemini research and technology”. In: *arXiv preprint arXiv:2403.08295* (2024).

[39] Nava Tintarev and Judith Masthoff. “A Survey of Explanations in Recommender Systems”. In: *2007 IEEE 23rd International Conference on Data Engineering Workshop*. 2007, pp. 801–810. DOI: `10.1109/ICDEW.2007.4401070`.

[40] Miles Turpin et al. “Language models don’t always say what they think: unfaithful explanations in chain-of-thought prompting”. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS ’23. New Orleans, LA, USA: Curran Associates Inc., 2024.

[41] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[42] Alexandra Vultureanu-Albişi and Costin Bădică. "Recommender Systems: An Explainable AI Perspective". In: *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2021, pp. 1–6. DOI: `10.1109/INISTA52262.2021.9548125`.

[43] J.H.M. Wagemans. *Argument Type Identification Procedure (ATIP) - Version 4*. English. Dec. 2021.

[44] J.H.M. Wagemans. "Constructing a Periodic Table of Arguments". English. In: *Argumentation, Objectivity and Bias: Proceedings of the Ontario Society for the Study of Argumentation Conference, Vol. 11, University of Windsor, May 18-21, 2016*. Ed. by L. Benacquista and P. Bondy. OSSA, 2016.

[45] J.H.M. Wagemans. *Periodic Table of Arguments 2.5 A*. English. May 2021.

[46] Yan Wang et al. *Enhancing Recommender Systems with Large Language Model Reasoning Graphs*. 2023. arXiv: `2308.10835 [cs.IR]`.

[47] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837.

[48] Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. "Measuring Association Between Labels and Free-Text Rationales". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 10266–10284. DOI: `10.18653/v1/2021.emnlp-main.804`. URL: `https://aclanthology.org/2021.emnlp-main.804`.

[49] Likang Wu et al. "A Survey on Large Language Models for Recommendation". In: *CoRR* abs/2305.19860 (2023).

[50] Likang Wu et al. "A survey on large language models for recommendation". In: *World Wide Web* 27.5 (2024), p. 60.

[51] Likang Wu et al. "A survey on large language models for recommendation". In: *World Wide Web* 27.5 (Aug. 2024). ISSN: 1573-1413. DOI: 10.1007/s11280-024-01291-2. URL: http://dx.doi.org/10.1007/s11280-024-01291-2.

[52] Evelyn Yee et al. "Faithful and Unfaithful Error Recovery in Chain of Thought". In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=IPZ28ZqD4I.

[53] Yongfeng Zhang and Xu Chen. "Explainable Recommendation: A Survey and New Perspectives". In: *Found. Trends Inf. Retr.* 14.1 (Mar. 2020), pp. 1–101. ISSN: 1554-0669. DOI: 10.1561/1500000066. URL: https://doi.org/10.1561/1500000066.

[54] Yongfeng Zhang et al. "Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis". In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '14. Gold Coast, Queensland, Australia: Association for Computing Machinery, 2014, pp. 83–92. ISBN: 9781450322577. DOI: 10.1145/2600428.2609579. URL: https://doi.org/10.1145/2600428.2609579.

[55] Yuhui Zhang et al. "Language Models as Recommender Systems: Evaluations and Limitations". In: *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*. 2021. URL: https://openreview.net/forum?id=hFx3fY7-m9b.

[56] Haiyan Zhao et al. "Explainability for Large Language Models: A Survey". In: *ACM Trans. Intell. Syst. Technol.* (Jan. 2024). Just Accepted. ISSN: 2157-6904. DOI: 10.1145/3639372. URL: https://doi.org/10.1145/3639372.

[57] Zihuai Zhao et al. *Recommender systems in the era of large language models (llms)*. 2024.

[58] Joyce Zhou and Thorsten Joachims. "GPT as a Baseline for Recommendation Explanation Texts". In: *Proceedings of the IntRS'23 Workshop*. 2023.

Appendix A

Prompts

A.1 Prompts for Explanations' Arguments Classification

Here, we present the prompt we used for obtaining the classification of the types of arguments present in the explanations sentences for the experiment of Chapter 4. We present it in text in this document, but the code that generates it is present in the experiment codebase[1].

PROMPT A.1.1: Explanations' Arguments Classification

```
Task: You will act as a Classifier. Your task is to identify
the type of argument present in each input sentence based on
the predefined classes below. Follow the instructions
carefully and output your results in the specified format.

CLASSES:

1. No Argument
   Code: NA
   Description: Sentences that do not present an argument.
   Example: "Anna is eating ice cream."

2. Argument from Analogy
   Code: An
```

[1] https://github.com/issilva5/llm-recommendations-survey

Description: Argues that because two things are similar, what is true of one is also true of the other.

Example: "Cycling on the grass is prohibited, because walking on the grass is prohibited."

3. Argument from Authority

Code: Au

Description: Relies on the opinion of an authority figure as evidence.

Example: "We only use 10\% of our brain, because Einstein said so."

4. Argument from Commitment

Code: Comm

Description: States a claim supported by something the addressee has previously said.

Example: "You said it yourself, we're the best in the world."

5. Argument from Comparison

Code: Comp

Description: Evaluates a situation, object, or idea by comparing it to another.

Example: "The president in wartime should not be swapped, because horses when crossing streams should not be swapped."

6. Argument from Criterion

Code: Cr

Description: Makes an argument considering specific criteria.

Example: "Haarlem is a better city to visit than Amsterdam because Haarlem has fewer tourists."

7. Argument from Disjuncts
   Code: Di
   Description: Concludes something happened because an alternative did not.
   Example: "He must have gone to the pub, because the interview is canceled."

8. Argument from Effect
   Code: Ef
   Description: Draws from consequences to infer the cause.
   Example: "In the centre of Amsterdam, parking rates must have gone up, because there are more empty spaces on the streets."

9. Argument from Equality
   Code: Eq
   Description: Based on the principle of equality, asserting that similar circumstances should lead to similar treatment.
   Example: "Dutch royals should pay taxes, because every other citizen pays taxes."

10. Argument from Evaluation
    Code: Ev
    Description: Argues based on personal experiences of good /bad or effective/ineffective.
    Example: "You should do paragliding because it is a great experience."

11. Argument from Opposites

```
    Code: Op
    Description: Concludes something by presenting the
persuasiveness of its opposite.
    Example: "Since false statements are persuasive, you
should believe the opposite too."

12. Argument from Sign
    Code: Sg
    Description: Asserts that the presence or absence of one
thing indicates the presence or absence of another.
    Example: "She likes Patricia because she is looking at
her all the time."

13. Argument from Similarity
    Code: Sim
    Description: Infers a fact based on its similarity to
another fact.
    Example: "Establishing gun control in present-day U.S.
will lead to genocide, because establishing gun control in
historical Germany led to genocide."

14. Argumentum ad Populum
    Code: Po
    Description: Asserts something is true or correct because
 many people think so.
    Example: "Many people said I'm the best writer in the
world, so I'm."

15. Pragmatic Argument
    Code: Pra
    Description: Evaluates actions, events, or rules based on
```

```
 their favorable or unfavorable consequences.
    Example: "Sleeping in the dark should be done by children
, because sleeping in the dark prevents them from ruining
their eyesight."

INPUT FORMAT

You will receive one or more sentences as input. Each
sentence represents a potential argument.

INPUT:
1. <sentence 1>
2. <sentence 2>
3. <sentence 3>

OUTPUT FORMAT

Your output should be a JSON object with the following
structure:

{
  "classification": [
    "Code_for_sentence_1",
    "Code_for_sentence_2",
    ...
  ]
}
```

A.2 Prompts for Faithfulness Experiment

Here, we present the prompts used to obtain the recommendations and explanations in the Faithfulness Experiment of Chapter 5. We present it in text in this document, but the code that generates it is present in the experiment codebase[2].

PROMPT A.2.1: Movies - Recommendation

```
You are a movie recommendation assistant. Your task is to
recommend movies to a user based based on the provided
information about the user preferences. The user wants two
movie recommendations that their will like (aka. positive)
and two movie recommendations that they will not (aka.
negative).


----------
Input:
- Liked movies: <liked_movie_1>; <liked_movie_2>; <
liked_movie_3>
- Disliked movies: <disliked_movie_1>; <disliked_movie_2>; <
disliked_movie_3>
----------
Output:
- The output must be a JSON object.
- Include no additional text besides the JSON
- The JSON object must have the following format:
    {
        "recommendations": [
            {
                "title": "movie title",
                "positive": true
            },
```

[2]https://github.com/issilva5/llm-explanations-faithfulness

```
            {
                "title": "movie title",
                "positive": true
            },
            {
                "title": "movie title",
                "positive": false
            },
            {
                "title": "movie title",
                "positive": false
            }
        ]
    }
```

PROMPT A.2.2: Movies - Explanation

```
You are a movie recommendation assistant. Your task is to,
based on the information provided about the user's
preferences, explain why they received a certain
recommendation. The recommendation can either be positive (a
movie they should watch) or negative (a movie they should
avoid).

----------
Input:
- Liked movies: <liked_movie_1>; <liked_movie_2>; <
liked_movie_3>
- Disliked movies: <disliked_movie_1>; <disliked_movie_2>; <
disliked_movie_3>

Recommendation: <recommendation>
```

```
Recommendation type: <positive|negative>
----------
Output:
- The output must be a JSON object.
- Include no additional text besides the JSON
- The JSON object must have the following format:
    {
        "explanation": "explanation"
    }
```

PROMPT A.2.3: Songs - Recommendation

```
You are a music playlist recommendation assistant. Your task
is to recommend songs to be added to a user’s playlist based
on the provided information about the playlist. The user
wants two songs that fit the playlist (aka. positive) and two
 songs that do not (aka. negative).

----------
Input:
- Playlist’s title: <playlist_title>
- Playlist’s current songs:
  1. <song_1> by <artist>
  2. <song_2> by <artist>
  3. <song_3> by <artist>
----------
Output:
- The output must be a JSON object.
- Include no additional text besides the JSON
- The JSON object must have the following format:
    {
        "recommendations": [
```

```
            {
                "title": "Song title",
                "artist": "Artist",
                "positive": true
            },
            {
                "title": "Song title",
                "artist": "Artist",
                "positive": true
            },
            {
                "title": "Song title",
                "artist": "Artist",
                "positive": false
            },
            {
                "title": "Song title",
                "artist": "Artist",
                "positive": false
            }
        ]
    }
```

PROMPT A.2.4: Songs - Explanation

```
You are a music playlist recommendation assistant. Your task
is to, based on the provided information about the playlist,
explain why they received a certain recommendation. The
recommendation can either be positive (a song they may add to
 the playslit) or negative (a song they should avoid).

----------
```

```
Input:
- Playlist's title: <playlist_title>
- Playlist's current songs:
  1. <song_1> by <artist>
  2. <song_2> by <artist>
  3. <song_3> by <artist>

Recommended Song: <recommendation_title> - by <
recommendation_artist>
Recommendation type: <positive|negative>
----------
Output:
- The output must be a JSON object.
- Include no additional text besides the JSON
- The JSON object must have the following format:
    {
        "explanation": "explanation"
    }
```

PROMPT A.2.5: Books - Recommendation

```
You are a book recommendation assistant. Your task is to
recommend books to a user based based on the provided
information about the user reading preferences. The user
wants two book recommendations that their will like (aka.
positive) and two book recommendations that they will not (
aka. negative).

----------
Input:
- Liked books: <liked_book_1> by <author>; <liked_book_2> by
<author>; <liked_book_3> by <author>
```

```
- Disliked books: <disliked_book_1> by <author>; <
disliked_book_2> by <author>; <disliked_book_3> by <author>
----------
Output:
- The output must be a JSON object.
- Include no additional text besides the JSON
- The JSON object must have the following format:
    {
        "recommendations": [
            {
                "title": "Book title",
                "author": "Author",
                "positive": true
            },
            {
                "title": "Book title",
                "author": "Author",
                "positive": true
            },
            {
                "title": "Book title",
                "author": "Author",
                "positive": false
            },
            {
                "title": "Book title",
                "author": "Author",
                "positive": false
            }
        ]
    }
```

PROMPT A.2.6: Books - Explanation

```
You are a book recommendation assistant. Your task is to,
based on the provided information about the user reading
preferences, explain why they received a certain
recommendation. The recommendation can either be positive (a
book they should read) or negative (a book they should avoid)
.

----------
Input:
- Liked books: <liked_book_1> by <author>; <liked_book_2> by
<author>; <liked_book_3> by <author>
- Disliked books: <disliked_book_1> by <author>; <
disliked_book_2> by <author>; <disliked_book_3> by <author>

Recommendation: <recommendation_title> - by <
recommendation_author>
Recommendation type: <positive|negative>
----------
Output:
- The output must be a JSON object.
- Include no additional text besides the JSON
- The JSON object must have the following format:
    {
        "explanation": "explanation"
    }
```

A.3 Prompts for Ranking Items based on the Explanation

Here, we present the prompts to obtain the rankings of the input items based on their importance to the recommendation as perceived in the explanation. We present it in text in this document, but the code that generates it is present in the experiment codebase[3].

PROMPT A.3.1: Prompt for Ranking Items in the Explanation for Movies

```
The movie <recomendation> was recommended to [not] be watched
 by a certain user.

The recommended system provided the following explanation: <
explanation>.

Your task is to given the recommendation and its explanation,
 provide a ranking of the input features based on their
importance to the recommendation.

----------
Input features:
- Liked movies: <liked_movie_1>; <liked_movie_2>; <
liked_movie_3>
- Disliked movies: <disliked_movie_1>; <disliked_movie_2>; <
disliked_movie_3>
----------
Output:
- The output must be a JSON object.
- The JSON must have one key, "ranking", which is a list of
the input features in order of importance.
- Each movie is an input feature. All features must be
included in the ranking.
- Consider the following feature names: ["liked_movie_1", "
liked_movie_2", "liked_movie_3", "disliked_movie_1", "
disliked_movie_2", "disliked_movie_3"]
- Include no additional text besides the JSON.
```

[3]https://github.com/issilva5/llm-explanations-faithfulness

```
- The JSON object must have the following format:
    {
        "ranking": [
            {"name": "feature_name_1", "value": "feature_1}",
            {"name": "feature_name_2", "value": "feature_2}",
            {"name": "feature_name_3", "value": "feature_3}",
            ...
        ]
    }
```

PROMPT A.3.2: Prompt for Ranking Items in the Explanation for Songs

```
The song <recomendation> was recommended to [not] be added to
 a user playlist.

The recommended system provided the following explanation: <
explanation>.

Your task is to given the recommendation and its explanation,
 provide a ranking of the input features based on their
importance to the recommendation.

----------
Input features:
- Playlist's title: <playlist_title>
- Playlist's current songs:
  1. <song_1> by <artist>
  2. <song_2> by <artist>
  3. <song_3> by <artist>
----------
Output:
- The output must be a JSON object.
```

```
- The JSON must have one key, "ranking", which is a list of
the input features in order of importance.
- The playlist’s title and each song are input features. All
features must be included in the ranking.
- Consider the following feature names: ["playlist_name", "
song_1", "song_2", "song_3", "song_4", "song_5"]
- Include no additional text besides the JSON.
- The JSON object must have the following format:
    {
        "ranking": [
            {"name": "feature_name_1", "value": "feature_1}",
            {"name": "feature_name_2", "value": "feature_2}",
            {"name": "feature_name_3", "value": "feature_3}",
            ...
        ]
    }
```

PROMPT A.3.3: Prompt for Ranking Items in the Explanation for Books

```
The book <recomendation> was recommended to [not] be read by
a certain user.

The recommended system provided the following explanation: <
explanation>.

Your task is to given the recommendation and its explanation,
 provide a ranking of the input features based on their
importance to the recommendation.

----------
Input features:
- Liked books: <liked_book_1>; <liked_book_2>; <liked_book_3>
```

```
- Disliked books: <disliked_book_1>; <disliked_book_2>; <
disliked_book_3>
----------
Output:
- The output must be a JSON object.
- The JSON must have one key, "ranking", which is a list of
the input features in order of importance.
- Each book is an input feature. All features must be
included in the ranking.
- Consider the following feature names: ["liked_book_1", "
liked_book_2", "liked_book_3", "disliked_book_1", "
disliked_book_2", "disliked_book_3"]
- Include no additional text besides the JSON.
- The JSON object must have the following format:
    {
        "ranking": [
            {"name": "feature_name_1", "value": "feature_1}",
            {"name": "feature_name_2", "value": "feature_2}",
            {"name": "feature_name_3", "value": "feature_3}",
            ...
        ]
    }
```