LightGCNxGPT: integrating LightGCN with GPT for enhanced personalised recommendations and explainability in recommender systems

Tiyyagura, Rochana

2024

Tiyyagura, R. (2024). LightGCNxGPT: integrating LightGCN with GPT for enhanced personalised recommendations and explainability in recommender systems. Final Year Project (FYP), Nanyang Technological University, Singapore.
https://hdl.handle.net/10356/175242

https://hdl.handle.net/10356/175242

SCSE23-0495

LightGCNxGPT: Integrating LightGCN with GPT for Enhanced Personalised Recommendations and Explainability in Recommender Systems

By

Tiyyagura Rochana (U2023234H)

Supervised By: Dr Liu Siyuan

Abstract

Recommendation systems have emerged as a pivotal tool in shaping our daily choices. Traditional systems face many challenges in providing users with accurate recommendations, especially when there is limited data. Furthermore, these systems fail to provide users with explanations for the recommendations made which is crucial in cultivating trust and transparency.

In light of the recent focus on Large Language Models (LLMs), this work proposes a novel framework called LightGCNxGPT that improves recommender systems by employing effective methods such as neighbourhood aggregation and user and item refinement. The LLM based paradigm proposed leverages upon the power of GPT, a popular LLM, to enhance the recommendations made by the state-of-the-art LightGCN model through innovative techniques, namely (i) User Information Refinement (ii) Item Noise Filtering (iii) GPT-Based Explanation Generation. Furthermore, theoretical analysis is provided to support the rationale behind the work and chosen methodology.

The experimental results evaluated on a benchmark dataset showcases that the LightGCNxGPT model **demonstrates superior** performance over current state-of-the-art models.

Acknowledgement

I would like to extend my heartfelt and sincere gratitude to my professor, Dr. Liu Siyuan and Dr. Zhou Xin, for their invaluable guidance, unwavering support, and expertise throughout the duration of my Final Year Project. Their encouragement and mentorship has been instrumental in shaping my understanding in this research area and I am very thankful for the opportunity to complete my project with them.

I would also like to express my appreciation to Nanyang Technological University for providing me with an enriching environment and opportunity to fulfil this project.

I would also like to extend my gratitude to my family and friends for their constant encouragement and support.

Table of Contents

List of Figures

1. Introduction

In recent times, recommendation systems have gained prominence, playing an important factor in our daily lives. They can powerfully influence our decision making processes across a multitude of domains, ranging from e-commerce to content streaming. However, traditional recommendation systems have proven to be limited in their performance and are unable to provide explanations for the recommendations made, which can be beneficial for users to understand how it relates to their needs and interests, as well as for developers to develop their model [1]. The current implementations of recommendation systems typically purely focus on content based methods and collaborative filtering based methods. The former utilises the features of items to find similarities with the user profile while the latter relies on user-item interactions, leading to user privacy concerns. These methods tend to offer limited personalization that does not capture users' distinct preferences. Furthermore, there is a notable deficiency in the diversity of recommendations, as these systems often prioritise popular or similar items, missing opportunities to introduce users to novel and fresh choices [2].

One of the foremost challenges faced by traditional recommenders is the 'Cold Start Problem,' wherein these systems struggle to provide relevant recommendations for new users or items lacking historical interaction data [3]. With the emergence of Large Language Models (LLMs), there is a promising avenue for mitigating the ‘cold start problem’ and improving the performance of recommenders. LLMs, especially Generative Pre-Trained Transformer (GPT), demonstrate exceptional prowess in comprehending textual and contextual information, without historical interaction data, as it can analyse user profiles and item descriptions through natural language processing. For instance, it can understand behavioural patterns or how temporal changes in popularity can affect user preferences [4], [5]. LLMs work well with Graph Convolutional Networks (GCNs) which exhibits proficiency in modelling intricate relationships and data structures to capture the item-item relationship presented on knowledge graphs [6], [7]. Through this synergy, recommendation systems can be characterised by heightened accuracy and the capacity to address a diverse spectrum of recommendations.

To bridge these gaps, we propose LightGCNxGPT, an enhanced and novel framework utilising GPT and GCNs, specifically improving upon the state-of-the-art GCN based recommender, LightGCN. The objective of this study is twofold. Firstly, by designing and implementing a system that harnesses the strengths of GPT and GCNs, we are able to capture the intrinsic relationship between users, context and items more accurately, hence improving the recommendations made. Secondly, by providing explanations for the recommendations made, we aim to cultivate transparency and trust between the uses and the system. This research presents the methodology used and the evaluation metrics along with a description of the experimental results. The scope of this research is limited to the use of GPT, but offers suggestions on extending the work to other LLMs. Through this research, we aspire to contribute to the advancement of recommendation systems, fostering enhanced user experiences and paving the path for tailored recommendations that can improve our daily lives.

1.1 Project Objectives

This Final Year Project aims to explore the integration of GCNs and GPT in Movie Recommendation Systems, focusing on the synergy between two advanced paradigms that can overcome the limitations of existing approaches. By delving into the capabilities of GCNs to model graph-based dependencies and GPT's prowess in contextual understanding, the research endeavours to present a more comprehensive and contextually aware recommendation system. It sets the platform for a novel approach that maximises the strengths of both GCNs and GPT.

In summary, the objective and contribution of the work presented is to showcase the improved performance of the LightGCNxGPT model against the current state-of-the-art recommender systems. LightGCNxGPT is the pioneering work that uses LLMs to refine the recommendations of its base model.

2. Literature Review

2.1 Traditional Methods

2.1.1 Collaborative-Filtering Recommendation Systems

Collaborative filtering provides recommendations on items based on explicit user ratings of items (users' historical data of ratings) [8]. It maintains a database of the user's preferences and finds the users' neighbours with whom they share similar preferences. It seeks to find similar users to the target user and provides suggestions based on the ratings and preferences of the similar users [9]. The similar users are identified via similarity calculation techniques such as Pearson correlation, cosine similarity, mean square difference, etc [10].

Collaborative filtering techniques is a popular approach as it does not require heavy domain knowledge, but rather a rating matrix to develop a factorization model [8]. However, it also has many challenges such as improving its scalability as modern systems are used by millions of users requiring the model to search through a larger space. Furthermore, based on the definition of user interaction, i.e., browsing patterns, CTR, rating, etc, a user with large amounts of information could reduce scalability as with large data points for a user the search for neighbours will slow down significantly. However, to improve the quality of these recommendations, one would require more data to provide more accurate recommendations. Hence, there is a trade-off between improving scalability and accuracy via these methods [9].

Collaborative filtering also does not perform well when faced in a 'cold start', i.e., the situation that arises when there is little to no information available to the model to make a recommendation [11].

2.1.2 Content-Based Recommendation Systems

Content based filtering methods provides recommendations by profiling a user to predict ratings on unseen items. As the predicted recommendations are user specific, content-based recommenders will not require other users' data, hence the system will be able to handle many users. However, recommenders face difficulty in providing accurate suggestions when there is not enough information of the user available. Furthermore, these systems recommend items based on their attributes and user preferences, leading to a limitation in providing novelty to the user – limitation arises because these systems rely on matching a user's profile to the features of the items. Thus, the recommenders tend to suggest items that tend to be similar to items the users have previously interacted with, limiting the diversity of suggestions and hindering the user from exploring new items [8], [12], [13].

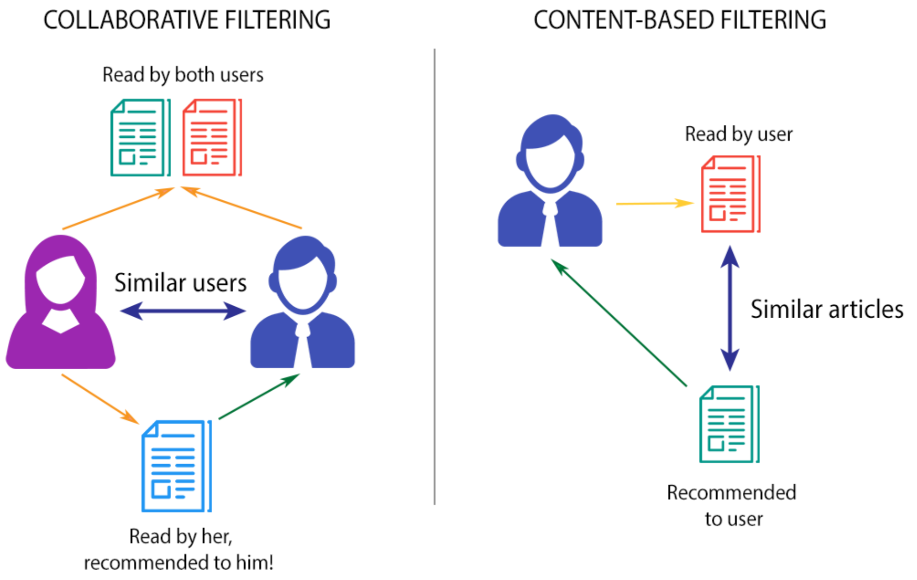

Figure 1. Collaborative Filtering vs Content-Based Filtering

2.2 Challenges faced by Recommender Systems

Due to the evolving and dynamic nature of human behaviour and tastes, consistently providing accurate recommenders prove to be a challenging task for recommender systems. Human tastes are ever evolving, influenced by a multitude of factors, such as personal experiences, trends, and changing preferences due to environmental factors or biological factors.

2.2.1 Cold-Start Problem

The cold start problem arises when the recommender systems struggle to provide accurate recommendations when there is little to no data available regarding a user or item with very limited data. For a new user, the challenge arises as there is very limited user historical data to accurately capture their like, preferences, and behaviour. These systems may instead recommend popular items or non-personalised suggestions leading to an unsatisfied user experience, discouraging their engagement with the platform, creating a circular problem.

2.2.2 Data Sparsity

Data sparsity occurs due to users generally ranking only a limited set of items. This leads to sparse user-item matrices, typically due to the lack of incentive to rank items [15]. Thus, there tends to be non-personalised or vague suggestions being made for users who provide less ratings as it is difficult for recommender systems to learn about the users' preferences.

2.2.3 Scalability

With the use of recommender systems on platforms with many users and/or items, scalability poses to be a pertinent issue. It can be difficult for to find a relevant neighbour for a certain user when there is many users and data.

2.2.4 Diversity of Recommendations

Typically, recommendation systems tend to provide users with suggestions like their previously interacted with items. At the same time, the most accurate results also occur when recommending items based on the users or objects' similarity. This is diversity issue limits a user to very similar suggestions and hinders them from discovering new and relatively niche items. A potential concern is that accuracy will be lost when there is diversity [16].

2.3 Evolution of Recommendation Systems

There has been a rise in hybrid methods that can be used to address the challenge outlined above. Recommendation systems have recently shifted to leverage more sophisticated approaches such as Graph Neural Networks (GNN) as they are able to model complex relationships in user-item interaction data. They utilise node features and graph structures to capture complex relationships [17]. They can overcome the issue of traditional methods that struggle to capture intricate patterns, especially in scenarios where the underlying data exhibits a graph-like structure. These networks operate on graphs where the nodes represent the users or items, and edges capture relationships. GNNs enable the aggregation of information from neighbouring nodes, facilitating a more comprehensive understanding of user-item interactions.

2.3.1 Graph Convolutional Network

Graph Convolutional Network (GCN) is a specific type of GNN that utilises convolutional process to learn patterns via graph structures [17]. It learns user and item interactions by linearly propagating them on the graph and uses the weighted sum of the embeddings learned at all layers as the final embedding [18]. GCNs has become the state-of-the-art for collaborative filtering and is widely used in recommender systems. However, it was found that the most common used techniques – nonlinear activation and feature transformation – makes it difficult for the model to train and degrades the performance of the model, as they produce little to the collaborative filtering performance [18].

GCNs and Collaborative filtering are the most fundamental technique used to cater personalised recommendations. Of the various approaches proposed, Wang et al. has proposed Neural Graph Collaborative filtering (NGCF) which has previously achieved state-of-the-art performance. This model utilises feature transformation, nonlinear activation, and neighbourhood aggregation to refine embeddings as inspired by GCNs. However, the LightGCN model as proposed in the paper by He et al. [18], has proven to be effective in providing better recommendations by enhancing graph convolutional networks and simplifying the focus of techniques to neighbourhood aggregation. This simpler model has proven to perform better as outlined in table 1– a notable 16% improvement - over NGCF, the state-of-the-art GCN based recommender. It is also a simpler model to train and implement into various systems [18].

Dataset	Gowalla		Yelp2018		Amazon-Book	
Method	**Recall**	**NDCG**	**Recall**	**NDCG**	**Recall**	**NDCG**
NGCF	0.1570	0.1327	0.0579	0.0477	0.0344	0.0263
Multi-VAE	0.1641	0.1335	0.0584	0.0450	0.0407	0.0315
GRMF	0.1477	0.1205	0.0571	0.0462	0.0354	0.0270
LightGCN	**0.1830**	**0.1554**	**0.0649**	**0.0530**	**0.0411**	**0.0315**

Table 1. Comparison of overall performance between LightGCN and State-of-the-Arts

It outperforms its competitors over all three datasets, namely Gowalla, Yelp2018, Amazon-Book. Hence, for this work, the chosen baseline model is the LightGCN model which has demonstrated superiority over the current state-of-the-art methods utilised in recommendation systems.

2.4. Light Graph Convolutional Networks (LightGCN)

LightGCN is a notable GNN model designed for recommendation tasks, belonging to the family of Graph Convolutional Networks (GCNs). It was developed by simplifying the design of the GCN and focusing only on the neighbourhood aggregation technique. LightGCN learns user and item 'embeddings' by linearly propagating them on the user-item interaction graph and uses the weighted sum of the embeddings learned at all layers as the final embedding [18].

For each user in the graph, within each layer a weighted sum of embeddings is computed from all the neighbourhood items. The equation used for calculation is

$$\mathbf{e}_u^{(k+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} \mathbf{e}_i^{(k)}$$

where k represents the layer and $e_u^{(k)}$, $e_i^{(k)}$ are the user, item node embeddings. $|N_u|$ and $|N_i|$ are the number of neighbours for the user and item nodes'. For each item, the weighted sum is similarly calculated by using the weighted sum of the neighbourhood users' using the below equation.

$$\mathbf{e}_i^{(k+1)} = \sum_{i \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} \mathbf{e}_u^{(k)}$$

where k represents the layer and $e_u^{(k)}$, $e_i^{(k)}$ are the user, item node embeddings. $|N_u|$ and $|N_i|$ are the number of neighbours for the user and item nodes'. The kth layer embeddings are derived after iterations on the nodes and is visualised as follows.

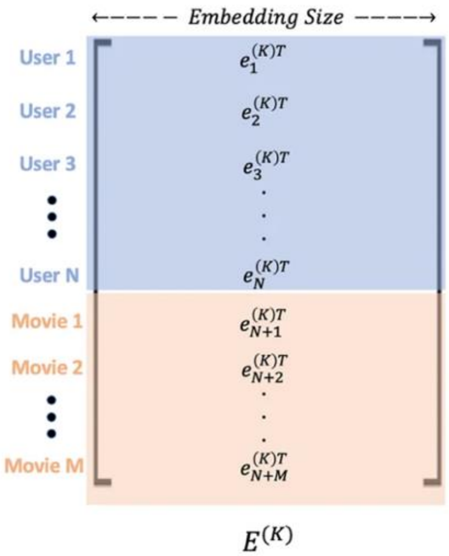

Figure 2. kth layer embedding of LightGCN Model

LightGCN then computes a weighted sum of the embeddings at all layers. The value of alpha (‘weights’) is a hyperparameter that can be learned or set.

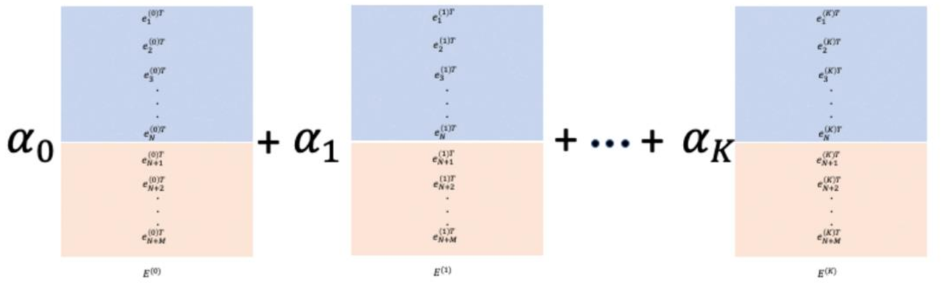

Figure 3. Combination of the Final Layer of LightGCN Model

The predictions are made based on the product of item and user embeddings using the equation.

$$\hat{y}_{ui} = \mathbf{e}_u^T \mathbf{e}_i$$

This product is indicative of the similarity between the user and the movie, hence becoming the basis for choosing an item for recommendation [19].

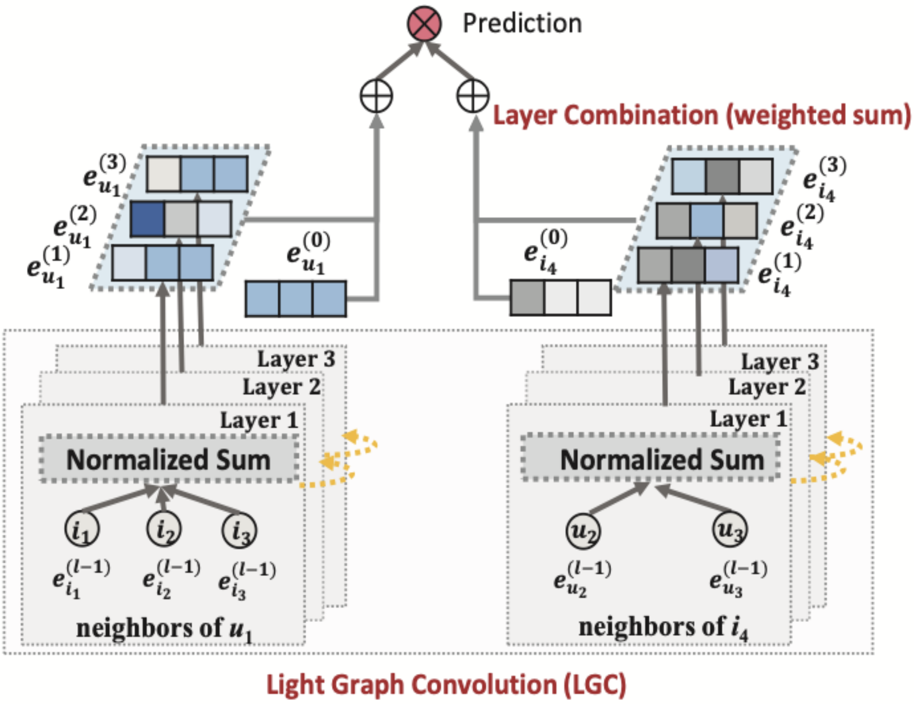

Figure 4. LightGCN Model Architecture

As illustrated in Figure 4, it uses a layer-wise propagation with only the normalised sum of the neighbour embeddings. At the layer combination, it sums the embeddings at each layer to obtain the final representations. It avoids the need for non-linear activation functions or feature transformation which leads to a more scalable and efficient model [18].

It also offers advantages as it can accurately and effectively enhance collaborative filtering and simplify the GCN architecture. It is also able to address the data sparsity issue due to its inherent ability to propagate information through the graph efficiently. Even in situations where explicit user-item interactions are limited, LightGCN can provide meaningful recommendations by leveraging graph-based dependencies.

The foundation of the graph construction for the LightGCN model lies in the symmetrically normalised adjacency matrix which represent the relationships between items and users in the graph. The matrix ensures a symmetric and scaled edge weights which facilitates a stable message passing process. The model also employs a multi-scale diffusion strategy that iteratively propagates information across the graph and passes messages between the nodes. This allows the model to aggregate information from the neighbourhood and refine node embeddings. Users and items are represented as low-dimensional vectors capturing their latent features. These embeddings are able to capture the characteristics of users and items in the graph. This is the mechanism that allows the model to capture both local and global dependencies, leading to enhanced performance as the learned embeddings can capture the underlying relationships. The structured approach enhances the model's ability to infer user preferences and make personalised predictions [18].

Many studies highlight the model's ability to outperform traditional collaborative filtering methods and state-of-the-art GNN architectures in terms of accuracy and computational efficiency, hence making it a good baseline model for the evaluation of our work and research goal.

2.5 Use of Large Language Models in Recommender Systems

The recent advancement of Large Language Models (LLMs) has sparked interest in its ability to refine machine learning frameworks. The integration of LLMs in recommender system has been a subject of focus recently, driving a shift towards more interactive and user based understanding. LLMs proves its ability in processing and generating natural language, allowing recommender systems to understand preferences better.

3. Methodology

3.1 Overview of the proposed approach

In this work, we propose a novel methodology that harnesses the power of Graph Neural Networks, specifically LightGCN, with Generative Pre-Trained Transformers (GPT) to enhance the recommendations made by recommendation systems. The primary goal of this project is to leverage upon GPT's natural language understanding coupled with LightGCN's model predictions via graph-based recommendation approaches. This integration of GPT into LightGCN's base model allows us to refine the predictions by filtering noise and utilizing contextual information extracted from user preferences to generate better and more accurate predictions. The integration of GPT with GCN not only aims to enhance the accuracy of recommendations but also aims to improve the interpretability of the recommendations made. By leveraging upon GPT's natural language generation capabilities, the system can offer personalised explanations for the recommendations made and allow users to gain insights into why specific items are recommended to them, carving the path of fostering transparency and trust between the users and the systems

The MovieLens dataset has been chosen as the main source of data for the training and evaluation of the recommendation system as it is a popular choice amongst benchmark datasets, especially in the field of recommendation systems.

3.2 Graph Construction using GCN

The graph construction involves the representation of users and items as nodes with edges representing the interactions between them. LightGCN in focus, builds a graph upon the concept of collaborative filtering where users and items are connected to each other via their interactions.

In the graph structure, ‘entities’ are represented as nodes and the ‘relationships’ between entities are described by edges. Users and movies are nodes on either side with edges connecting them, describing their relationships indicating user-movie interactions. An edge exists if a user rates a movie with a higher rating, taken as rating >= 4.0, and there will be no edge if a user rates it lower than the defined threshold of 4.0 [19].

3.3 GPT as an LLM

GPT as an LLM for refining recommendations of the LightGCN model presents a pioneering approach in the realm of recommender systems. Its natural language processing capabilities allows it to capture and understand the context and nuances of user preferences and pick up on trends/patterns.

Leveraging the reasoning and knowledge based abilities of GPT, our methodology seeks to profile users and refine item recommendations by filtering for ‘noise’.

3.3.1 GPT Prompt Engineering

In the context of refining LightGCN model’s recommendations, GPT prompt engineering plays a critical role in our chosen research methodology. This involves the meticulous designing of prompts to effectively leverage upon GPT’s natural language understanding. The aim is to create prompts that aids GPT in providing insightful and more accurate recommendations. Our objective is to ensure that GPT aligns to the nuanced requirements of movie recommender systems, firstly by filtering out ‘noise’ or irrelevant suggestions and recommending movies that are catered and resonate with the user’s profile and interests. The prompts used in this work has been inspired by studies [20], [21], that have demonstrated the effectiveness of careful prompt engineering in eliciting high-quality outputs, and through brute experimentation of different prompts and comparing their results.

User Information Refinement

Utilizing GPT for user-profiling on limited information of purely the movies they have previously liked can offer a wealth of inferred information such as the user's demographic and preferences. This involves GPT analysing the history of liked movies and identifying patterns that align with the user. For example, a preference for a certain director or genre can hint at a broader understanding of the user. It also utilizes the information of the user's liked history to make educated guesses on the demographics of the user, such as their age, gender, and country of residence. This mitigates privacy concerns as users can limit the amount of personal information they offer on these platforms and still enjoy better quality recommendations. We are also able to utilize the reasoning ability of GPT to further refine recommendations made by the LightGCN model.

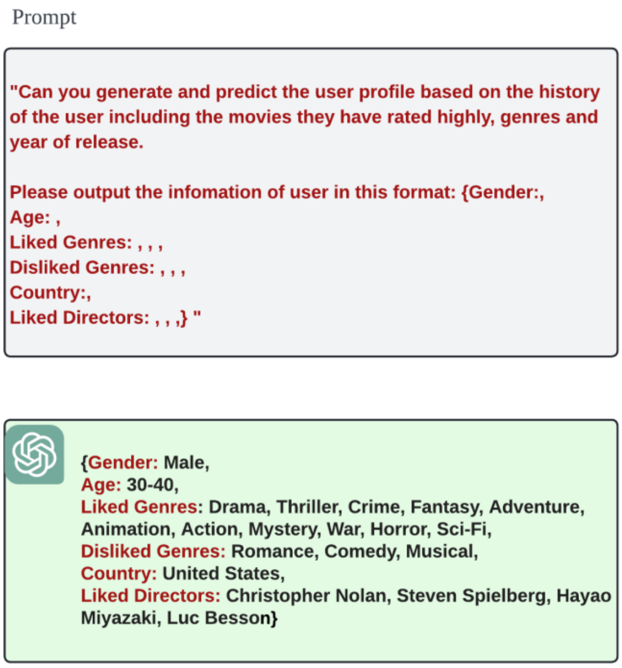

Figure 5. User Information Refinement GPT Prompt

This additional inferred information allows the model to refine recommendations, ensuring they are not purely based on the explicit preferences of the users (i.e. the liked movie history) but also able to integrate the implicit characteristics deduced.

Refining Recommendations – Item Noise Filtering

In term of refining recommendations, GPT plays an important role in filtering out movies that does not align with the user preferences, despite it being popular or highly rated by other users. GPT's natural language processing ability allows it to assess the context and subtleties of the user's liked history, allowing it to discern why specific movies are not specific, hence termed 'noise', and removes these from the suggestions.

> **movies_removed = ["The Matrix (1999)", "The Sixth Sense (1999)", "Fight Club (1999)"]**
>
> **Explanation:**
>
> **- These movies have been removed from the recommendations as they do not align closely with the user's stated movie preferences and genre interests. While they are highly acclaimed films, they may not match the specific criteria derived from the user's liking patterns and genre choices.**

Figure 6. GPT Item Noise Filtering

As seen in Figure 6, GPT is able to filter out 'noise' despite it being a popular choice amongst users as it did not fit the profile of the user, which is a feature most recommender systems are unable to do.

Refining Recommendations – Item Predictions

GPT can recommend other movies it deems to be more suitable for the user based on the comprehensive learned user profile and identifying movies with similar themes, styles, doctorial signatures to those the user has showed preference to.

Explanation:

1. **"The Green Mile (1999)":** This movie is recommended because it **shares similar themes** of redemption, justice, and emotional depth as "The Shawshank Redemption". It is a captivating drama with strong performances and a compelling storyline.

2. **"Gladiator (2000)":** This epic historical drama **aligns** with the **user's preference** for action and drama. It features intense battle scenes, a gripping story of vengeance and loyalty, and strong character development, similar to "Braveheart".

3. **"Inglourious Basterds (2009)": Quentin Tarantino's** unique blend of action, drama, and historical fiction makes this a fitting recommendation for someone who enjoyed "Terminator 2: Judgment Day". It is a stylized and engaging film with memorable characters and unexpected plot twists.

4. **"The Departed (2006)":** This crime thriller is recommended due to its **intense storyline,** moral complexity, and stellar cast, reminiscent of "Saving Private Ryan". The movie keeps viewers on the edge of their seats with its suspenseful plot and dynamic characters.

5. **"Black Hawk Down (2001)":** With its gritty portrayal of modern warfare and emphasis on camaraderie and sacrifice, this action-packed movie should **appeal to fans of "Apollo 13"**. It delivers intense military action sequences and explores themes of bravery and resilience.

Figure 7. GPT Item recommendations and rationale

As seen in Figure 7, GPT is able to understand implicit preferences and recommend movies on reasons such as choice of directors, commonality of themes, storyline, etc, and does not limit to one-dimensional suggestions.

This dual capability of GPT – to filter out and enrich in – significantly enhances the relevance and quality of recommendations provided, making the model more user-centric and powerful.

3.3.2 Explanation Generation using GPT

Explainability in recommendation systems is paramount to building trust between the users and the system as it provides them a clear understanding and explanation behind the suggested items. This will likely lead to users being more trusting and willing to accept the recommendations. As GPT is able to produce coherent and contextually relevant explanations, it can prove itself to be powerful in bridging the gap between the algorithmic decision-making process by the Machine Learning Model and user comprehension. It mitigates any potential misunderstanding that could arise due to vague recommenders. Furthermore, these explanations carry its own set of advantages in allowing a user the flexibility of choosing between suggested items based on their current mood and needs, especially since it is difficult for machine learning models to pick up on a users' current mood/wants immediately.

GPT Explanation Generation

1. "**Prisoners**"

Explanation: "Prisoners" is a gripping thriller that delves into complex moral dilemmas, similar to "**Inception**" and "The Shawshank Redemption." The tense atmosphere, strong performances, and intricate storytelling are likely to appeal to your preference for suspenseful movies.

2. **"Arrival"**

Explanation: As a fan of science fiction like "**Interstellar**," "**Arrival**" offers a unique take on extraterrestrial communication and time perception. The film's thought-provoking narrative and Amy Adams' stellar performance make it a must-watch for sci-fi enthusiasts.

3. **"Gone Girl"**

Explanation: "Gone Girl" combines elements of thriller and drama, providing a twisted and suspenseful story that keeps viewers on the edge of their seats. If you **enjoyed the tension and psychological depth** in "The Shawshank Redemption," this movie is likely to captivate you.

Figure 8. Example Response of GPT Explanation Generation for recommendations made

GPT Explanation Generation

1. **"La La Land"**
Explanation: Since you enjoyed movies with a **mix of romance and music genres**, "La La Land" is a great recommendation for you. It is a musical film that received critical acclaim for its performances and music.

2. "**Arrival**"
Explanation: Given your interest in thought-provoking and sci-fi films, "Arrival" is a compelling choice. This movie offers a unique take on the science fiction genre with its focus on communication and time.

3. "**Moonlight**"
Explanation: Since you **appreciate character-driven** dramas and emotional storytelling, "Moonlight" is likely to resonate with you. The film explores themes of identity and self-discovery in a poignant and visually striking way.

Figure 9. Example Response of GPT Explanation Generation for recommendations made

Many recommender systems fail to provide explanations to users which often leads to a sense of uncertainty and distrust with the system, drawing user retention lower. If a user is provided with the rationale behind suggestions also allows the user to refine their preferences and provide feedback to the system, further improving the accuracy and quality of personalised recommendations over time.

Depending on the domain of the recommendation system, explanations can prove to have a significant impact, especially prevalent in e-commerce or healthcare related platforms. This will allow users to make more informed decisions and build trust with the system.

The specific implementation and prompt used in this work is further elaborated in section 4.5.

3.4 Integration of LightGCN and GPT

In our proposed approach, the integration of Light Graph Convolutional Network (LightGCN) empowered with Generative Pre-Trained Transformers (GPT) promises a novel and innovative approach of enhancing the recommendations systems by offering better recommendations and personalised explanations. The LightGCN model forms the backbone of the recommender system, utilising its graph-based collaborative filtering techniques to accurately capture the user-item interactions and generated more accurate recommendations. GPT is known for its power in understanding and generating natural language and hence acts complimentary to LightGCN.

By providing prompts to GPT based on LightGCN's model predictions and information about a user and movies they have previously rated highly (such as the movies titles, genres, and year release), we harness GPT's power in contextual understanding to improve the recommendations as well as provide tailored explanations, improving the users' experience with the recommender system.

This integration of two powerful frameworks, paves a path towards a novel approach of refining recommendation systems. The combination of graph-based collaborative filtering and natural language generation delivers personalised and interpretable recommendations, fostering user engagement and trust.

In summary, our GPT-based paradigm involves:

- **User Information Refinement:** Using GPT prompt to generate additional user information that were not originally present in the dataset. This additional information is fed into GPT to further refine recommendations based on the user profile and preferences.
- **Item Noise Filtering:** Using GPT prompt to refine recommendations by removing and adding movies to the base model's recommendations.
- **GPT-Based Explanations**: Providing explanations for movie recommendations to the user.

4. Experiment Design

4.1 Dataset Description

The dataset chosen for this project is the MovieLens 100k dataset, a widely chosen benchmark dataset.

Key Characteristics of the dataset:

- **100,000 Movie Ratings:** The daaset comprises of a total of 100,000 ratings provided by all users for various movies
- **609 Users:** The user pool consists of 609 users and their interactions with differences movies (ratings)
- **9,742 Movies:** The dataset includes 1682 unique movies released from a range of genres

Each user included in the dataset has rated at least 20 movies.

4.1.1 Exploratory Data Analysis

The MovieLens dataset consists of two primary datasets, namely the movie dataset and the rating dataset. The movie dataset encompasses information about the movies, including its movieId, title, and genre. A movie can be defined through multiple genres, providing more insightful information.

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

Figure 10. MovieLens Movie Dataset

The rating dataset consists of user interactions represented via the userId, movieId, ratings, and timestamps. Users can provide ratings for a movie on a scale of 0.5 to 5.0 for the movies they have watched.

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

Figure 11. MovieLens Rating Dataset

Through exploratory data analysis, we can observe that a majority of users tend to leave ratings to rate a movie highly. In figure 12, we can observe that there are more user ratings for movies with higher average rating. Possibly, users who like a movie are more likely to engage with the platform by leaving a rating, hence there will be more ratings left for the upper rating scales.

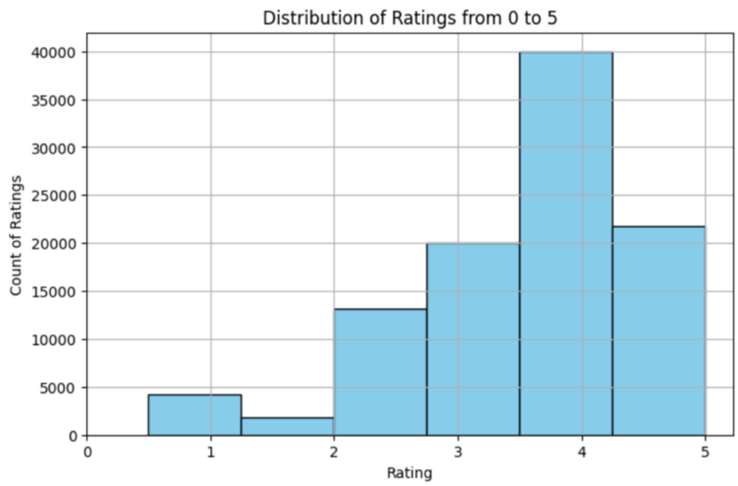

Figure 12. Distribution count of user ratings

In Figure 13, we can observe that an average user tends to leave very few ratings which makes it difficult for systems to profile the user based on their tastes, hence hindering the model from accurately capturing their interests, leading to more popular and generic movies being suggested.

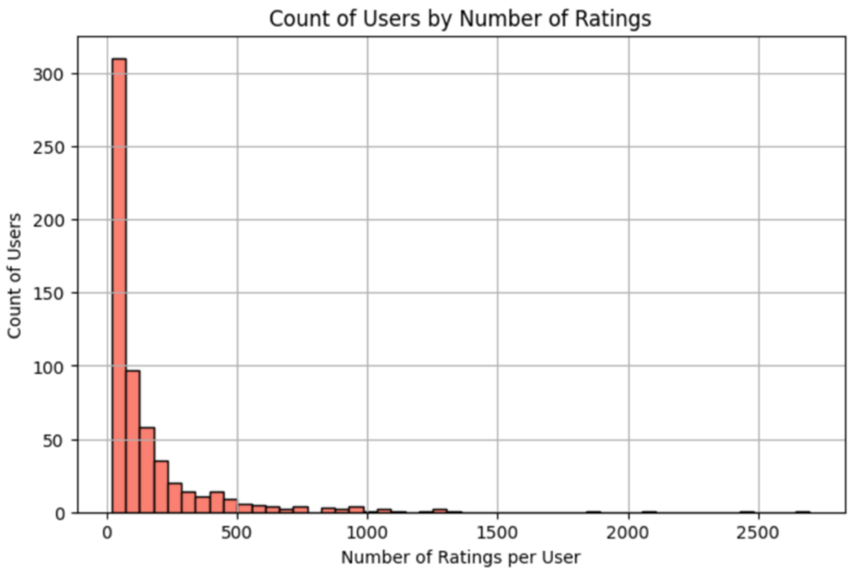

Figure 13. Count of Users vs Number of ratings

Upon making these observations, we can say that the MovieLens dataset is suitable for this project as it emulates the real-world use case of the model. In most recommendation platforms, we observe that an average user will leave less ratings and may only feel like interacting with the platform when they liked the item. Hence, this dataset allows us to frame our project in a way that tackles the real-world challenges described in section 2.2. By using this dataset, we can evaluate how our improved model can tackle challenges such as data sparsity and the cold start problem.

4.2 Evaluation Metrics

In evaluating the performance of the recommendation system, it is important to employ appropriate metrics that are indicative of the quality and effectiveness of the recommendations provided. The popular key metrics used are Precision, Recall, and

F1 Score. The ground truth of the user refers to all the items ('movies') the user has highly rated (rating >= 4.0).

Precision@k

Precision refers to the proportion of correctly recommended items among the total recommended items, hence conveying the accuracy of recommendations. It is the key metric in determining the performance of our recommendation system as our goal is to evaluate the how many relevant recommendations have been made within the top k items.

$$Precision\ = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Specifically, we use these to calculate the metrics on the TopK recommended items, i.e. precision@k.

$$Precision@K\ = \frac{Number\ of\ Relevant\ Items\ in\ Top\ K}{K}$$

Our goal is to maximise the relevance of recommendations presented to the users, hence making precision a key metric to evaluate the performance of our model by. Furthermore, from a scaling perspective, precision indicates a more accurate depiction of the performance as there is a constraint on how many recommendations can be provided to the user, due to their attention span and memory requirements. For example, if the system can only provide 50 recommendations, we seek to be as precise with the chosen recommendations [22].

Recall@k

Recall measures the proportion of relevant items that were recommended to users among all relevant items. While it provides some valuable insights, it introduces bias in the evaluation as users have varying preferences and their total number of relevant items can differ across users. Hence, it may potentially skew the results and not accurately reflect the performance of the model [22].

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Specifically, we use these to calculate the metrics on the TopK recommended items, i.e. recall@k.

$$Recall@K = \frac{Number\ of\ Relevant\ Items\ in\ Top\ K}{Total\ Number\ of\ Relevant\ Items}$$

In the case of recommendation systems, recall can be calculated as the number of movies recommended that were present in the ground truth for each user.

4.3 Model Implementation

4.3.1 Preprocessing and Training

To ensure the suitability and adaptability of raw data for evaluating the performance of the model, the data was pre-processed and transformed into a clean and structured format to be used on a GCN. The pre-processing phase involves loading the user and movie data ('nodes') from their CSV files and establishing edges between the users and movies based on their ratings. The dataset is then split using a 60:20:20 ratio to ensure sufficient data for the model training and evaluation. It is an appropriate choice for this project as it facilities the machine learning model training, hyperparameter tuning, avoiding overfitting, and unbiased evaluation, thus producing a recommender system that performs well on real world data.

Random sampling prepares mini batches of interactions for training the data. It involves selecting a batch size of interactions, consisting of the user, positive rated items, negatively sampled items to balance the representation and reduce bias. The positive edges refer to user-item interactions that indicate that the user liked a certain item or has interacted with it, whereas a negative edge indicates that the user did not like it or interact. To achieve this, we use a python package, *structured_negative_sampling*, to select a negative edge for every positive edge in the graph, in a manner that ensures that the dataset is balanced and helps in the training process to differentiate between positive and negative interactions. This ensures that the model is trained on the users' preferences and non-preferences to allow it to learn more robust and generalised patterns.

Based on prior literature review, the hyperparameters values chosen are as follows:

ITERATIONS	10,000
BATCH_SIZE	1024
LEARNING RATE	1e-3
ITERATIONS_PER_EVAL	200
ITERATIONS_PER_LR_DECAY	200
LAMBDA	1e-6

Table 2. Hyperparameters for LightGCN Model Training

The training process consisted of 10,000 iterations, each iteration consisting of a batch size of 1024 data points with iterative updates occurring with a learning rate of 1e-3. The interval of evaluation was set to every 200 iterations, accompanied with learning rate decay to enhance convergence. A lambda value of 1e-6 was applied to the training to prevent overfitting.

4.3.2 Loss Function

We utilsie the Bayesian Personalised Ranking (BPR) loss function, a pairwise function that prioritizes observed ('positive') user-item interactions to have higher values than unobserved interactions along with L_2 regularization, λ [19], [24].

$$L_{BPR} = -\sum_{u=1}^{M} \sum_{i \in N_u} \sum_{j \notin N_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda ||\mathbf{E}^{(0)}||^2$$

4.4 Implementation Details

The experiment was conducted on Google Colab Pro GPU using PyTorch for code implementation. The OpenAI GPT model used is GPT-3.5-turbo-0125, due to bigger requests size per day and for easy reproducibility of this work.

4.5 Experimental Setup

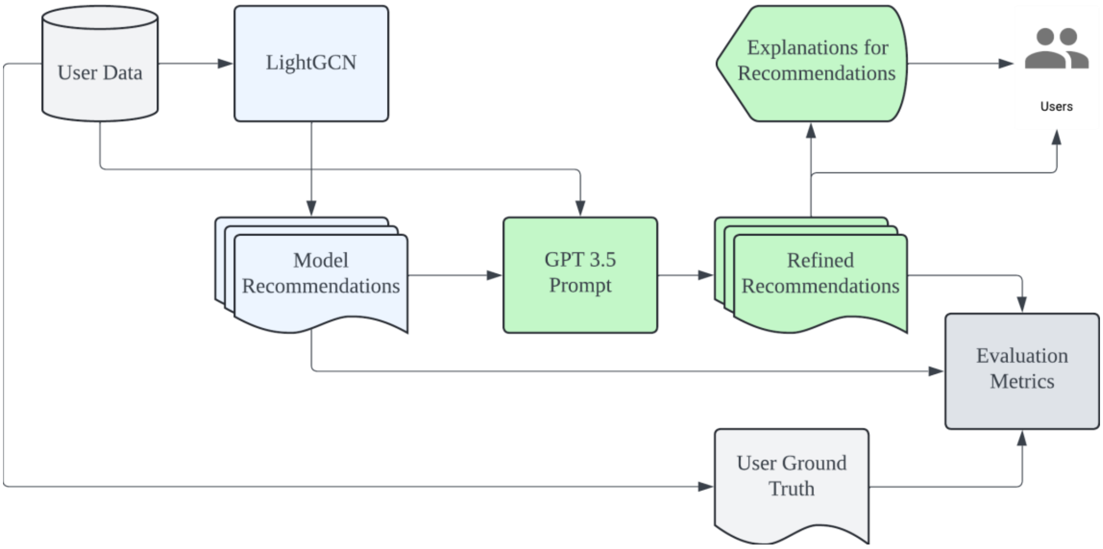

Figure 14. Experimental Setup of LightGCNxGPT

In our experimental setup, we seek to compare the results of the recommendations provided by the base LightGCN model and the enhanced approach of using LightGCN with GPT, referred to as LightGCNxGPT.

First the LightGCN model is trained on the user data and leverages upon its graph-based collaborative filtering approach to capture intricate relationships between the user and items. Through its iterative message passing mechanism, the model is able to recommend movies based on its learned embeddings and suggest a list of top k movies for each user. Once the model predicts movies for the user, the recommendations are passed into GPT via a prompt for further refinement and enhancement. As GPT can understand context and generate natural language, it is able to provide personalised explanations for the refined recommendations made. Through prompt engineering and fine-tuning, GPT is able to refine the recommendations, enhancing the accuracy and relevance of recommendations made.

The sample architecture is as follows:

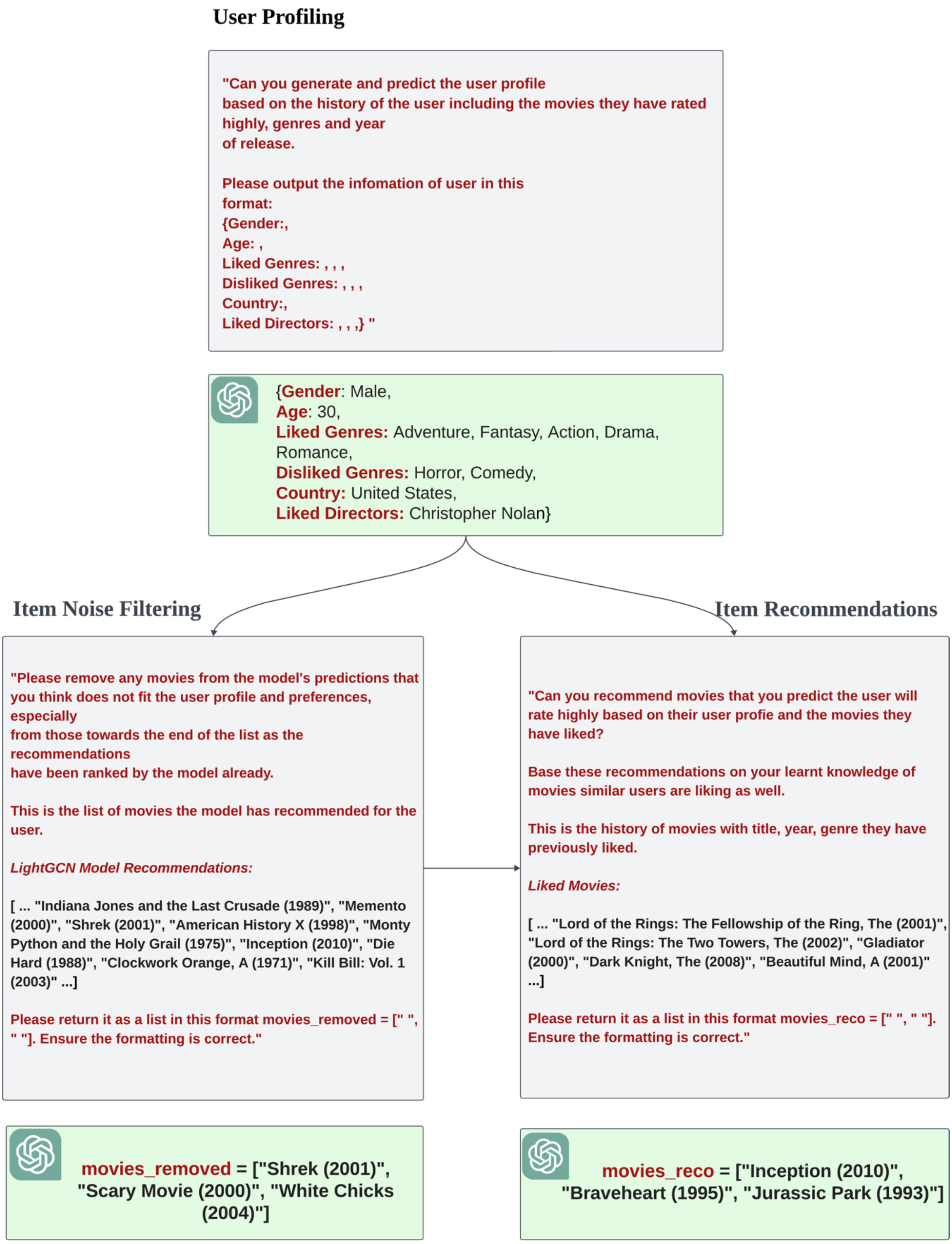

Figure 15. GPT Prompts used with example responses

Explanations

"For every movie recommended, can you provide an explanation for each movie as to why it has been suggested?

Be as specific as possible such that users are able to understand the recommendations better."

Explanation For All Recommended Movies:

1. **"Inception (2010)"**: Since you enjoyed "Dark Knight, The," directed by Christopher Nolan, and appreciate complex and thought-provoking narratives, "Inception" is a highly recommended
film. It blends elements of action and drama while exploring themes
of dreams and reality.

2. **"Braveheart (1995)":** Given your appreciation for epic and dramatic films like "Gladiator," "Braveheart" offers a similar experience. It features themes of bravery, loyalty, and historical conflicts, which align well with your taste in action and drama genres.

3. "**Jurassic Park (1993)**": As you enjoy adventure and fantasy genres such as "Lord of the Rings" and have shown an interest in visually stunning and thrilling films, "Jurassic Park" is a classic recommendation. It combines elements of action, suspense, and awe-inspiring visuals that could resonate with your movie preferences.

{explanations for all movies follows}

Figure 16. GPT Explanation Generation for refined movie recommendation list

The prompt and flow of data is as listed below:

- The user data, specifically the list of movies they have previously rated highly, is fed to GPT to analyse user preferences and tastes.

- GPT is then prompted to generate a user profile, inferring additional details about the user such as their age, gender, demographic data, etc.
- Once the user has been profiled, GPT will aim to refine the recommendations made by the LightGCN model by filtering out and enriching in movies.
- The refined list of recommendations is coupled with GPT's explanations for user interpretability.

4.6 Calculation of Metrics

Once GPT produces the list of refined recommendations, the scores of both the baseline LightGCN model and enhanced LightGCNxGPT model are computed and compared. The evaluation metrics used are precision and recall. For the calculation of precision, the percentage of top k suggested items intersecting with the user's test ground truth is computed. The user's ground truth is the positive items in the test data, i.e. items they have positively rated. Similarly, for recall, the number of positive items recommended in the top k is computed as a percentage of the user's ground truth. F1 Score is derived from the precision and recall scores.

5. Results

Through the systematic comparison and analysis of the two models on the MovieLens dataset, the LightGCNxGPT model has demonstrated superior performance.

Metric Scores (Averages)		**LightGCN**	**LightGCNxGPT**
TopK@10	Recall	0.0333	**0.0363 (+9%)**
	Precision	0.1822	**0.2032 (+11.5%)**
TopK@20	Recall	0.0530	**0.0631 (+19.1%)**
	Precision	0.1519	**0.1791 (+17.9%)**
TopK@30	Recall	0.0674	**0.0794 (+17.8%)**
	Precision	0.1335	**0.1574 (+17.9%)**
TopK@40	Recall	0.0804	**0.0900 (+11.9%)**
	Precision	0.1195	**0.1364 (+14.1%)**
TopK@50	Recall	0.0906	**0.1004 (+10.8%)**

	Precision	0.1069	**0.1197 (+11.97%)**

Table 3. Comparison of Performance of LightGCN and LightGCNxGPT Model

5.1 Discussion of Results

We performed detailed comparisons of results from both models, recording their respective performances at different values of TopK as well as the percentage of relative increase of recall and precision from the base model.

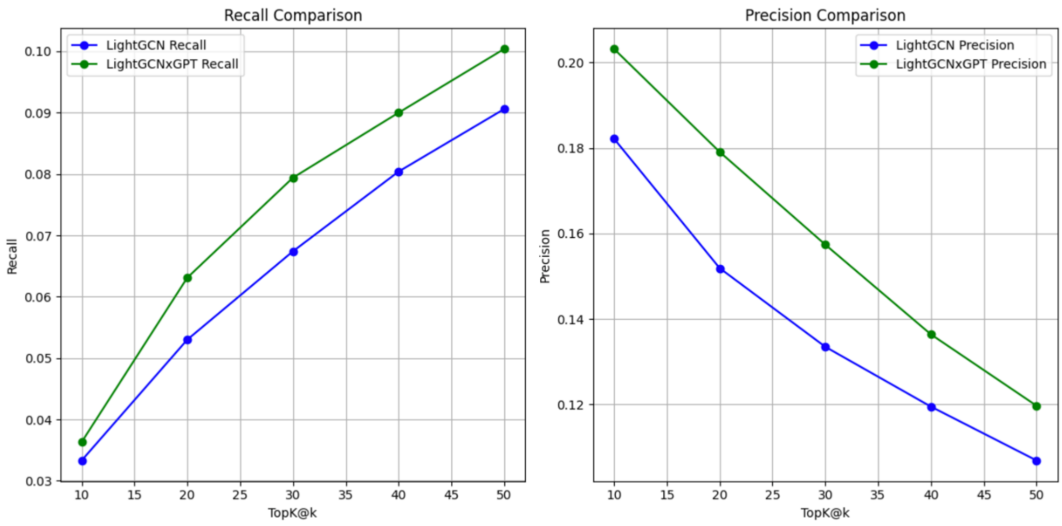

Figure 17. Recall and Precision Comparison of LightGCN and LightGCNxGPT

Key Observations:

- In all cases of TopK values, the **LightGCNxGPT outperforms** the LightGCN base model.
- Observably the **precision**, the key metric of evaluation, of the LightGCNxGPT model **improves significantly**. This means that the model is able to capture more relevant recommendations within the top k suggestions, which is a defining feature of an efficient recommendation system.

- The LightGCNxGPT model demonstrates a **more significant rate of increase** in recall across all TopK values. This highlights the models' ability to learn faster and capture more 'ground truths' in a shorter period.

Upon further evaluation on individual users, we can observe that LightGCNxGPT acted on par or better than the LightGCN model **over 92% of the users**. Hence, over 92% of users were provided with better recommendations as compared to the LightGCN model.

	Num Users with improved performance	**Total Num Users**	**% of Users with improved performance**
TopK@10	507	594	85.35
TopK@20	544	589	92.36
TopK@30	544	587	92.67
TopK@40	560	587	95.40
TopK@50	565	587	96.25
		Average	**92.4**

Table 3. Count of users with LightGCNxGPT Performance Improvement

5.1.1 Extension of Results

Empirically we have observed the increased performance of the LightGCNxGPT model amongst the fixed values of ‘k’. However, during the experimentation, we observe that the actual improvement of the model LightGCNxGPT may be slightly under-represented due to the size limitations of k. For instance, even though the model is able to predict relevant items fairly well, the performance of the models will only increase if GPT is able to remove movies it believes were ‘noise’ or wrongly predicted by the model and recommend a ground truth value within its first few suggestions. Furthermore, different users will have different number of rated movies as observed in Figure 13, causing the models and GPT to be exposed to different amounts of information about each user. As an extension of the previous results discussed, an interesting extension would be to calculate the scores of recall and precision for the recommendations made by both the LightGCN and LightGCNxGPT model for individual users without a fixed k value across all users. The rationale behind calculating this metric is to understand the models’ ability to predict relevant items without the constraint of ‘k. To ensure the fairness and comparability of the two models, we ensure that the size of recommended movie ids of both models for each user was the same, i.e., the smaller size of both models was taken as ‘TopK’ metric for calculation of recall and precision for each individual user.

	LightGCN	LightGCNxGPT
Average Recall	0.0577	**0.0691 (+19.8)**
Average Precision	0.1463	**0.1697 (+16.0%)**

Table 4. Comparison of Performance between LightGCN and LightGCNxGPT

We observe that the LightGCNxGPT model performs much better than the LightGCN model.

5.2 Limitations

Using GPT for movie recommendation systems carries its own set of limitations despite offering a wide range of advantages. One significant limitation is its occasional failure to respond and listen to prompts, even when explicit instructions are provided. For instance, when GPT is asked to provide the user with new movie recommendations, i.e. movies the user has not rated previously, the model recommends a previously rated movie despite explicit instructions. We approximate this issue to affect 8% of users during our experimentation. While this behaviour is not ideal, in the context of movie recommendations, recommending a movie the user has already watched could be appreciated by the user and is not a detrimental issue. However, these limitations highlight the need for further refinement in the training and implementation of GPT. While it has proven its natural language processing abilities, addressing these limitations will allow for a broaden applicability across various use cases. This calls for further research and development efforts to improve the performance of GPT based recommendation systems.

5.3 Future Work

To further the scope of this research work, we aim to apply our novel approach to different datasets to evaluate the performance of the LightGCNxGPT model and compare the results with other state-of-the-art models.

As for the scope of this project, it would be interesting to integrate the timestamp data to evaluate the performance of the models over real-time data and understand whether the models are able to pick up on evolving tastes or trends. In theory, GPT is able to understand the evolution of human behaviour and analyse how global trends can affect movie recommendations for users, which makes this an interesting extension to the project.

As a next step, we aim to develop a platform which integrates the LightGCNxGPT model to serve as a platform where users can get movie recommendations. Implementing such a platform will also allow for features such as user feedback that can be used for the further refinement of the model. Overall, the development of a platform serves as an exciting next step in evaluating the full performance of the model and delivering value to the users. We also aim to use other LLMs and evaluate the performance as compared to GPT.

6. Conclusion

In conclusion, this project has showcased the superior performance of LightGCNxGPT, a novel framework that combines the concepts of Graph Convolutional Networks (GCN) and Large Language Models (LLMs). Through meticulous experimentation and evaluation, the effectiveness and potential of the model in providing more accurate and personalised movie recommendations has been shown. By harnessing the power of collaborative filtering and natural language processing, LightGCNxGPT is able to enhanace the recommendation process and offer tailored suggestions. The results obtained in this work signifies a good step forward in the realm of recommendation systems. Ultimately this work has reached the research objective and has contributed towards the broader goal of enhancing user experience in recommendation systems, paving the path for more intelligent systems in the future.

Appendix

The code implementation of this work can be found here.

(https://github.com/Rochanaaa/LightGCNxGPT)

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

References

[1] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System." arXiv, Apr. 03, 2023. doi: 10.48550/arXiv.2303.14524.

[2] C. Gao *et al.*, "A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions." arXiv, Jan. 12, 2023. doi: 10.48550/arXiv.2109.12843.

[3] "Facing the cold start problem in recommender systems." Accessed: Mar. 25, 2024. [Online]. Available: https://www.researchgate.net/publication/261961547_Facing_the_cold_start_problem_in_recommender_systems

[4] W. Fan *et al.*, "Recommender Systems in the Era of Large Language Models (LLMs)." arXiv, Aug. 05, 2023. doi: 10.48550/arXiv.2307.02046.

[5] W.-C. Kang *et al.*, "Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction." arXiv, May 10, 2023. doi: 10.48550/arXiv.2305.06474.

[6] Z. Chen *et al.*, "Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs." arXiv, Jan. 16, 2024. Accessed: Mar. 25, 2024. [Online]. Available: http://arxiv.org/abs/2307.03393

[7] C. hao *et al.*, "LKPNR: LLM and KG for Personalized News Recommendation Framework." arXiv, Aug. 23, 2023. doi: 10.48550/arXiv.2308.12028.

[8] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Appl. Sci.*, vol. 10, no. 21, Art. no. 21, Jan. 2020, doi: 10.3390/app10217748.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, in WWW '01. New York, NY, USA: Association for Computing Machinery, Apr. 2001, pp. 285–295. doi: 10.1145/371920.372071.

[10] S. Gong and G. Cheng, “Mining User Interest Change for Improving Collaborative Filtering,” in *2008 Second International Symposium on Intelligent Information Technology Application*, Dec. 2008, pp. 24–27. doi: 10.1109/IITA.2008.385.

[11] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, “Content-Based Video Recommendation System Based on Stylistic Visual Features,” *J. Data Semant.*, vol. 5, pp. 1–15, Jun. 2016, doi: 10.1007/s13740-016-0060-9.

[12] S. Gong and G. Cheng, “Mining User Interest Change for Improving Collaborative Filtering,” in *2008 Second International Symposium on Intelligent Information Technology Application*, Dec. 2008, pp. 24–27. doi: 10.1109/IITA.2008.385.

[13] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., Berlin, Heidelberg: Springer, 2007, pp. 291–324. doi: 10.1007/978-3-540-72079-9_9.

[14] Yao-Ting(Teddy), “Implicit Feedback Recommendation System (I) — Intro and datasets EDA,” Medium. Accessed: Mar. 25, 2024. [Online]. Available: https://medium.com/@teddywang0202/implicit-feedback-recommendation-system-i-intro-and-datasets-eda-eda16764602a

[15] Department of Computer Science, Al Nahrain University, Baghdad-Iraq., N. F. Al-Bakri, S. H. Hashim, and Department of Computer Science, University of Technology, Baghdad-Iraq., “Reducing Data Sparsity in Recommender Systems,” *J. Al-Nahrain Univ. Sci.*, vol. 21, no. 2, pp. 138–147, Jun. 2018, doi: 10.22401/JNUS.21.2.20.

[16] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, “Solving the apparent diversity-accuracy dilemma of recommender systems,” *Proc. Natl. Acad. Sci.*, vol. 107, no. 10, pp. 4511–4515, Mar. 2010, doi: 10.1073/pnas.1000488107.

[17] T. Vaj, “GNN vs GCN vs GAN (Grap,” Medium. Accessed: Mar. 25, 2024. [Online]. Available: https://vtiya.medium.com/gnn-vs-gcn-vs-gan-grap-cf7a790848ad

[18] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation.” arXiv, Jul. 07, 2020. Accessed: Mar. 23, 2024. [Online]. Available: http://arxiv.org/abs/2002.02126

[19] Q. Wang, “LightGCN for Movie Recommendation,” Stanford CS224W GraphML Tutorials. Accessed: Mar. 24, 2024. [Online]. Available: https://medium.com/stanford-cs224w/lightgcn-for-movie-recommendation-eb6d112f1e8

[20] “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications.” Accessed: Mar. 25, 2024. [Online]. Available: https://arxiv.org/html/2402.07927v1

[21] “The More Is Not the Merrier: Effects of Prompt Engineering on the Quality of Ideas Generated By GPT-3.” Accessed: Mar. 25, 2024. [Online]. Available: https://www.researchgate.net/publication/374942989_The_More_Is_Not_the_Merrier_Effects_of_Prompt_Engineering_on_the_Quality_of_Ideas_Generated_By_GPT-3

[22] P. Aher, “Evaluation Metrics for Recommendation Systems — An Overview,” Medium. Accessed: Mar. 24, 2024. [Online]. Available: https://towardsdatascience.com/evaluation-metrics-for-recommendation-systems-an-overview-71290690ecba

[23] “Understanding and Applying F1 Score: AI Evaluation Essentials with Hands-On Coding Example,” Arize AI. Accessed: Mar. 24, 2024. [Online]. Available: https://arize.com/blog-course/f1-score/

[24] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian Personalized Ranking from Implicit Feedback,” 2009.