Personalized Movie Recommendation System on Douban Based on Deep Learning

Kanyu Bao
School of Computer Science and Technology
Harbin Institute of Technology, Shenzhen

Chen Lu(Teammate)
School of Computer Science and Technology
Harbin Institute of Technology, Shenzhen

Abstract

This project conducts social network analysis and personalized movie recommendation on the Douban platform. We visualize two distinct social networks within the Douban website, uncovering various network characteristics of the Douban community. Collaborative filtering methods are implemented and enhanced. Additionally, we implement two deep learning models: a Transformer-based model and a GPT-2 based model, and an improved GPT-2 based model. Our findings indicate that the improved GPT-2 based model outperforms the others in both the recommendation and explanation tasks.

1 Introduction

1.1 Background

With the popularization of the Internet, people's entertainment options have become increasingly diverse, and movies have become an essential part of daily life. However, given the vast number of available films, how can video platforms recommend suitable movies to users in a personalized manner, and how can users find movies they will enjoy? More importantly, how can these recommendations be provided with compelling justifications? To address these issues, our group has implemented a personalized recommendation system using the Douban movie dataset.

Douban Movies, recognized as the most authoritative and high-quality movie rating website in China, has amassed a vast collection of movies along with user ratings and reviews. This extensive dataset significantly aids in the development of our recommendation system. Additionally, the social networks formed within Douban can enhance the efficacy of the recommender system.

1.2 Dataset

The primary data source for this study is the open dataset available at http://moviedata.csuldw.com/, collected in early September 2019. This dataset comprises 140,502 movie records, 639,125 user records, and 4,428,475 comment records. In comparison to the dataset available on e-learning platforms, this dataset offers more extensive information, enabling us to incorporate additional features into the recommendation system to enhance its effectiveness.

Given the large volume of records and significant noise present in the original data, we performed data cleaning and selected a subset for analysis. Specifically, we chose movies with the top 10,000 comments and randomly selected 1,000 users. The comments were filtered to include only those related to the selected movies and users, resulting in a refined dataset consisting of 82,894 comment records. This curated dataset is saved as $reviews\_selected.csv$ and was used for our analysis.

Regarding the Douban user-fan relationship social network, we have constructed two datasets. The first dataset is sourced from an open-source repository on GitHub. This dataset consists of 1,000 users and 589 follow relationships, randomly crawled from the Douban website.

2024 Social Network Analysis (Xutao Li).

For the second dataset, we selected a subset of users from the initial dataset as seed nodes and employed our custom crawler to obtain their follow relationships. This resulted in a network comprising 1,000 users and 15,432 follow relationships, making it significantly denser than the first dataset.

2 Social Network Analysis and Visualization

2.1 Overview

In the dataset comprising the top 10,000 most popular movies, we define popularity as the frequency with which comments about each movie appear in the comment subset. The visualization of this metric is presented in Figure 1. Subsequently, we generated a word cloud of movie-related tags, and the visualization is displayed in Figure 2.

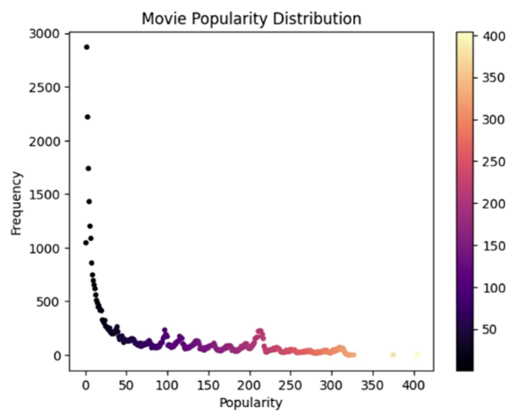


Figure 1: The observation result shows that the popularity of movies satisfies the Zipf distribution, fewer movies have higher popularity

Figure 2: Word cloud shows that tag $America$ appears the most frequently, which reflects users' love for American movies

2.2 Random Selected Social Network

The specific parameters of Douban user-fan relationship network crawled randomly are as follows:

Table 1: **specific parameters**

Parameters	number
nodes	1000
edges	589
density	0.016%
average path length	3.957

We then visualize it through Gephi, and the results is displayed in Figure 3 .

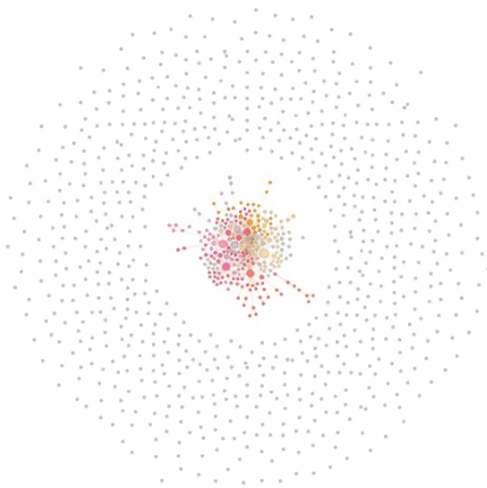


Figure 3: Random Selected Social Network

Observational Findings:

- The analysis reveals the presence of numerous discrete nodes, indicating that Douban does not function as a traditional social networking site like QQ or Weibo.
- The size of the nodes corresponds to their degree, with only a few nodes exhibiting a larger size.
- The color of the nodes denotes their respective communities, which were identified using the Fast Unfolding algorithm. Detailed information is provided below.

The Fast Unfolding algorithm, also known as the Louvain algorithm, is an iterative method designed to partition a network into communities with the objective of maximizing the network's modularity post-division. As described by Blondel et al.[1], the algorithm operates by iteratively refining the community structure to enhance the overall modularity. Modularity, a key measure used to evaluate the quality of the division, is defined as follows:

$$Q = \frac{1}{2m} \sum i, j \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

The algorithm process is primarily divided into two stages:

- The first stage, known as Modularity Optimization, involves assigning each node to the community where its neighboring nodes are located. This process is iteratively performed to maximize the modularity value.

- The second stage, referred to as Community Aggregation, entails merging the communities identified in the first stage into single nodes and reconstructing the network. This iterative process continues until there are no further changes in the network's structure.

After removing the discrete nodes, we employ two layouts, Force Atlas and Force Directed Layout, to visualize the central community within the network, as illustrated in Figure 4. The Force Atlas layout results in a more concentrated arrangement of nodes, providing an intuitive representation of the division of different communities within the network. In contrast, the Force Directed Layout more effectively illustrates the distances between nodes.

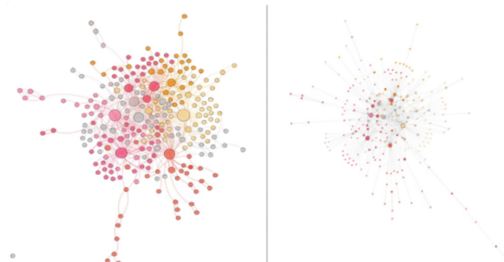


Figure 4: Central Group

2.3 Seed-Nodes-Started Social Network

Using the algorithm described above, we visualized the second social network with Gephi. The resulting visualization, presented in Figure 5, demonstrates the division of the network into seven communities, achieving a modularity score of 0.348.

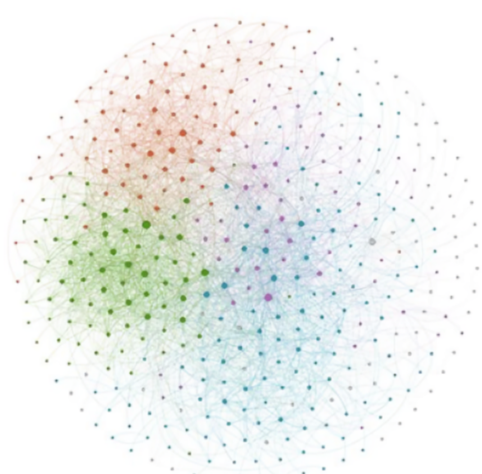


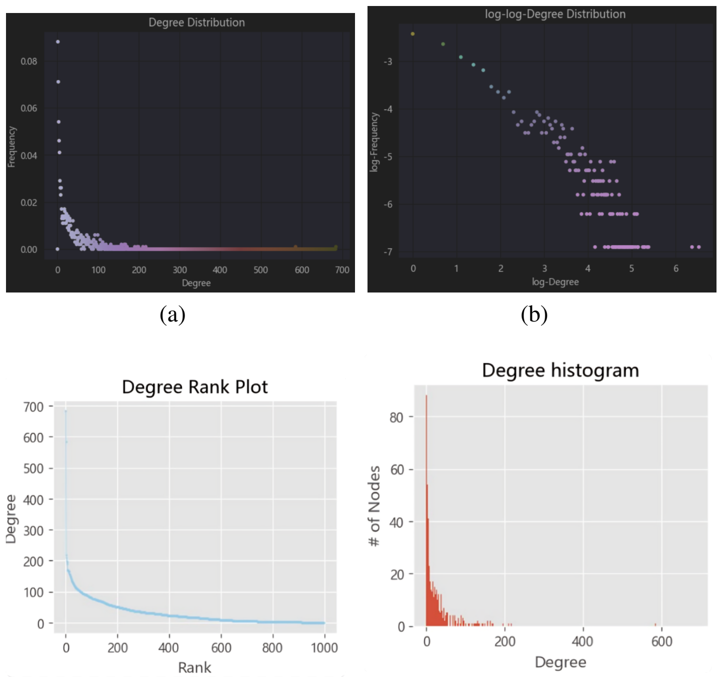
Figure 5: Seed-Nodes-Started Social Network

The specific parameters of the social network are as follows:

Table 2: **specific parameters**

Parameters	number
nodes	999
edges	15432
average degree	15.446
density	0.015
diameter	7
average path length	2.976

Next, we visualize some parameters of the network. The results are displayed in Figures 6, 7.



(a) (b)

(c) (d)

Figure 6: Degree distribution

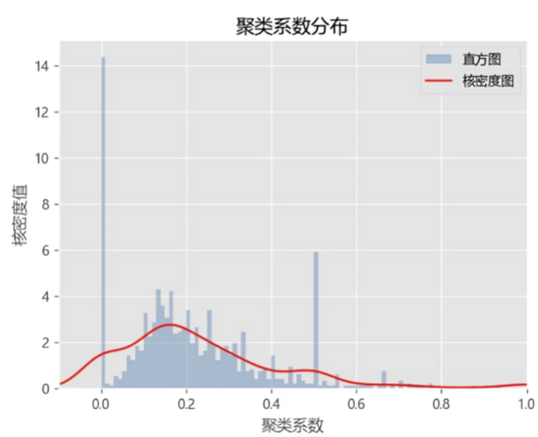


Figure 7: Clustering Coefficient

The degree and clustering coefficient also follow a Zipf distribution.

2.4 Centrality Measurement

We utilized the NetworkX library to calculate six types of centrality within the network. The visualization outcomes are depicted in Figure 9, where the color coding represents the magnitude of centrality: red indicates centrality values exceeding 50% of the maximum value, blue indicates values greater than 30%, and gray represents values less than 30%.

Our analysis reveals that different centrality measurements yield notably varied results, with the majority of nodes appearing gray in the first five visualizations.

Subsequently, we selected nodes randomly to measure degree and PageRank centralities using Gephi, extracting data for the top five users. The outcomes are illustrated in Figure 8.

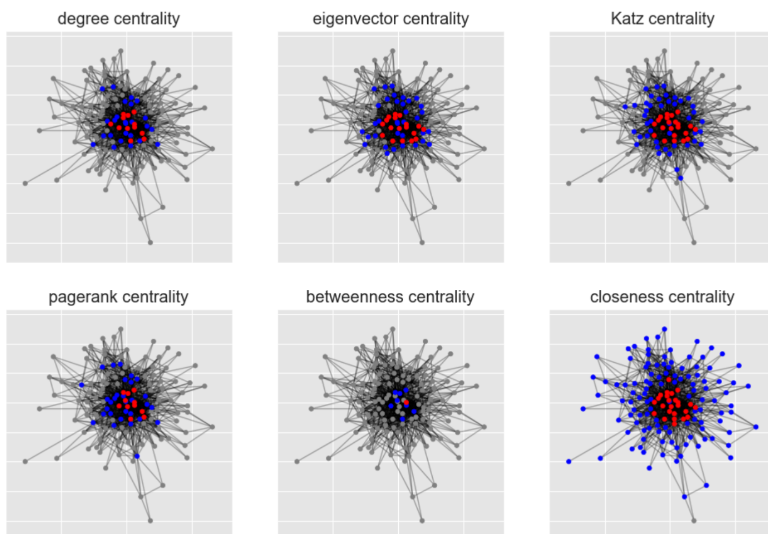


Figure 8: 6 types of centrality measurements

The results from these two measurement algorithms are remarkably consistent, both in terms of node distribution and the identification of the top five users.

3 Collaborative Filtering

In collaborative filtering, the principle hinges on homogeneity, where users exhibiting similar purchasing behaviors are presumed to have similar interests in items. Consequently, users tend to select items with comparable sales records. Given a dataset comprising M users, N items, and their corresponding sales or rating records represented by an $M \times N$ matrix R, User-based Collaborative Filtering (UCF) computes the similarity between any two users (u_a and u_b) based on their purchasing behaviors. This method predicts user u's behavior on item i as a weighted average of other users' behaviors on item j, with weights determined by the similarity between i and another user o. Conversely, Item-based Collaborative Filtering (ICF) computes the similarity between any two items (i_a and i_b) based on their sales records and predicts user u's behavior on item i using other items that user u has purchased.

Given that the rating record in the test dataset is discrete, ranging from 1 to 5, Pearson correlation coefficient is chosen as the similarity measure as it can distinguish between preferences (like and dislike), unlike cosine similarity. The formulation for UCF is depicted in Equation 1, while ICF follows a similar approach.

Given that the rating records from the test dataset are discrete values ranging from 1 to 5, we opted to use the Pearson correlation coefficient as the measure of similarity. This choice is based on its ability to differentiate between like and dislike, a distinction that cosine similarity fails to make. The application of User-based Collaborative Filtering (UCF) is demonstrated as follows, with Item-based Collaborative Filtering (ICF) following a similar approach.

$$sim(u_a, u_b) = \frac{\sum_k (r_{ak} - \bar{r}_a)(r_{bk} - \bar{r}_b)}{\sqrt{\sum_k (r_{ak} - \bar{r}_a)^2 \sum_k (r_{bk} - \bar{r}_b)^2}}$$

The computational complexity of User-Based Collaborative Filtering (UCF) is $M^2\times$N, while that of Item-Based Collaborative Filtering (ICF) is M$\times N^2$. Consequently, on platforms like Douban, where users significantly outnumber items and their preferences are relatively personal and stable, UCF tends to perform better. Conversely, ICF is more suitable for news platforms where preferences are more socially driven and items are updated more frequently. Additionally, UCF can integrate users' social information, thereby enhancing the explainability of recommendations, a topic that will be further discussed in the next section. For these reasons, we implemented both ICF and UCF in our dataset.

Collaborative Filtering (CF) faces several common challenges, such as the impact of sparsity on accuracy, the limitations of the cold start problem, and a lack of explainability. The cold start problem also affects evaluation—if the training set lacks all purchasing behavior of a user, their ratings are uniformly predicted as 3. Similarly, if the training set lacks all sales records of an item, its rating from each user is predicted as the user's average rating, and each predicted rating contributes to the final RMSE/MAE if it appears in the test set.

To mitigate these issues, we introduced several improvements. ICF can be enhanced with social network information, while UCF can be augmented with knowledge graph information. These two improved algorithms, referred to as ICFP and UCFP respectively, will be elaborated upon in the following sections.

3.1 ICFP

We have evaluated the impact of a socially bounded Item Collaborative Filtering (ICF) recommendation system, which only considers similar users within the user's immediate network. However, due to the sparsity of both the rating records and the social network, the number of usable references is further reduced, resulting in unsatisfactory outcomes.

In our subsequent attempt, we introduced node similarity within the social network as an additional dimension. This approach allows us to define user similarity as a weighted average of behavioral similarity and node similarity. Nonetheless, the results remained suboptimal. A possible explanation for this could be that, on platforms like Douban, many social connections are not based on similar film preferences but rather on general experience sharing. As a result, considering all connections may introduce noise into the similarity matrix.

However, on such platforms, users with high centrality might be senior film critics who provide more valuable information. A user may follow a senior film critic because they are interested in most of the films reviewed by the critic and find the reviews objective (though this objectivity is subjective). Therefore, we can identify a high-centrality node set H and measure the similarity between two users u_a and u_b based on the overlap of their followed high-centrality nodes using Jaccard similarity, as shown as follows. Moreover, when predicting a user's rating, the weight of a similar user who is also a high-centrality neighbor can be further increased.

$$sim_H(u_a, u_b) = \frac{|N(a) \cap N(b) \cap H|}{|N(a) \cup N(b) \cap H|}$$

3.2 UCFP

In the UCFP framework, we integrated User Collaborative Filtering (UCF) with a knowledge graph, concentrating on three principal attributes of a movie: directors, actors, and genres. The similarity for each attribute is computed according to follows, as delineated in Xiao et al.[2].

$$sim_p(i_a, i_b) = \frac{2|S_p(a) \cap S_p(b)|}{|S_p(a)| + |S_p(b)|}$$

The method for integrating the three similarity measures warrants further exploration. Our approach involves adaptively assigning weights based on the user's profile, with different users receiving different weights. This is because some users may prioritize the director, others may be ardent fans of specific actors, and still others may have a strong preference for certain genres.

To build a user's profile, we identify a range of entities and calculate the user's affinity for each entity t, as defined as follows. Here,$\overline{r_t}$ represents the average rating the user gives to items associated with t, and $freq_t$ denotes the proportion of items related to t in the user's purchasing history. This measure helps prevent the erroneous classification of a user as a fan of t if they happen to highly rate a single item related to t by chance.

$$A_t = \bar{r}_t + 5\sqrt{freq_t}$$

The variance of A_t within a property, as depicted in Equation 5, quantifies the extent of an individual's concern for that property. Consequently, V_p can be utilized as the weighting factor for sim_p when making predictions for this user.

$$V_p = Var_{t \in p}(A_t)$$

3.3 Result

As shown in Table 3, the performance of User-based Collaborative Filtering (UCF) is unsatisfactory due to data sparsity, and the User-based Collaborative Filtering with Popularity (UCFP) algorithm does not significantly improve UCF in terms of precision. The limited information provided by the sparse social network further contributes to this issue. However, the recommendation results, specifically the top 5 predicted items, become more explainable. Item-based Collaborative Filtering (ICF) outperforms UCF, indicating that preferences on this platform are more individualistic than social. Moreover, the Item-based Collaborative Filtering with Popularity (ICFP) algorithm enhances ICF by delving deeper into the similarity between items while also considering the differences among users.

Table 3: **Result**

Alg	RMSE	MAE
UCF	1.0066	0.78
UCFP	0.9365	0.7372
ICF	0.9193	0.7427
ICFP	0.8098	0.6902

4 Deep Learning Models

In this section, we present three deep learning models (Transformer-based, GPT2-based, and improved GPT2-based) designed for the Explainable Recommendation task. The first two models are adapted from PETER[3] and PEPLER[4] respectively, while the third model is proposed by us, demonstrating superior performance on this task. Additionally, to enhance the efficiency of these models on the Douban dataset, we introduce a natural language processing (NLP) pre-processing method aimed at extracting valuable information from user reviews.

4.1 Problem Formulation

Given a user u and an item i, our objective is to estimate a rating $\hat{r}_{u,i}$ that predicts u's preference towards i. Additionally, our model is designed to generate a natural language sentence $\hat{E}_{u,i}$ for the user-item pair, providing a justification for why i is recommended to u. To enhance the precision of both the recommendation and its explanation, we incorporate features $F_{u,i}$, which include the movie's tags, genres, director, actors, region, and release date. During the training phase, the explanation $E_{u,i}$ is provided as input. In the generation phase, we replace this input with <eos> to enable the model to autonomously generate the explanation.

4.2 NLP Pre-Process

Given the substantial noise present in the raw data, it is essential to extract the most informative explanations $E_{u,i}$ from each review. To achieve this, we will identify the top three most valuable sentences as the explanation. Our selection criteria prioritize sentences that exhibit two key characteristics: they reflect the user's preference, and they are closely related to the fundamental information about the movie. For example, a sentence like "Cameron's special effects are attractive" would be a suitable reason to recommend the movie "Avatar". To facilitate this process, we derive two scores. The similarity score score_sim assesses the similarity between the review sentence s and the movie introduction document D. The movie introduction, written by the movie producer, encapsulates the basic information about the movie. A higher similarity indicates that the sentence is more likely to contain essential information about the movie. We compute score_sim using the BM25 algorithm:

$$\text{score\_sim(s)=BM25(s, D)} = \sum_i IDF(q_i) \frac{f(q_i,D)(k_1+1)}{f(q_i,D)+k_1(1-b+b\frac{\|D\|}{avgdl})}$$

In this context, let q_1, q_2, $\cdots$, q_n represent the words contained in the sentence s. The function $f(q_i, D)$ denotes the frequency of the word q_i within the document D. The term $\|D\|$ refers to the total number of words in the document D, while avgdl stands for the average document length across the entire text collection from which the documents are sourced. The parameters k_1 and b are considered hyperparameters within this framework.

The $score\_sentiment$ metric evaluates how well a sentence aligns with the user's sentiment. For this purpose, we utilize the SnowNLP sentiment rating, which employs a trained Bayesian classifier. When a sentence s is entered, the system scans for all emotional keywords and uses the Bayesian classifier to assign a sentiment score, $sentiment(s)$, within the interval of 0 to 1. A higher $sentiment(s)$ value indicates a more positive emotion. Subsequently, the $score\_sentiment$ is computed as follows.

$$score\_sentiment(s) = -\|sentiment(s) * 5 - rating\|$$

In this context, the variable $rating$ represents the user's rating in the review, which ranges from 1 to 5, indicating the user's preference towards a movie. A higher sentiment score suggests a greater alignment with the user's sentiment. For each sentence s within a review, we combine the sentiment score and the rating to derive a total score. We then select the top three sentences based on these combined scores to form the explanation $E_{u,i}$:

$$score(s) = score\_sim(s) + score\_sentiment(s)$$
$$E_{u,i} = concatenate(s_1, s_2, s_3)$$
$$where \;\; score(s_1) \geq score(s_2)... \geq score(s_n)$$

4.3 Transformer based model

In this section, we present the details of our transformer-based model, which is adapted from PETER[3], originally introduced by Vaswani et al.[5]. First, we explain the encoding process for different types of tokens within a sequence. Next, we introduce the model's framework and the revised attention masking matrix. Finally, we formulate three tasks—explanation generation, context prediction, and recommendation—and integrate them into a multi-task learning framework.

As shown in Fig 9, the input to our model is a sequence consisting of user ID u, item ID i, features $F_{u,i}$, and explanation $E_{u,i}$. The input sequence can be represented as $S =< u, i, f_1, \cdots, f_{|F_{u,i}|}, e_1, \cdots, e_{|E_{u,i}|} >$, where $S =< f_1, f_2, \cdots, f_{|F_{u,i}|} >$ are the features and $< e_1, e_2, \cdots, e_{|F_{u,i}|} >$ are the words in the explanation sequence. Here, $|F_{u,i}|$ denotes the number of features, and $|E_{u,i}|$ denotes the number of words in the explanation. Clearly, there are three types of tokens in the sequence S, namely users, items, and words (including features). For these, we prepare three sets of randomly initialized token embeddings: U for users, I for items, and V for words. Additionally, we include positional embeddings P to encode the position of each token in the sequence. After performing embedding lookup and positional encoding, we obtain the input sequence $S_0 =< s_{0,1}, s_{0,2}, \cdots, s_{0,|S|} >$, where $|S|$ is the length of the sequence.

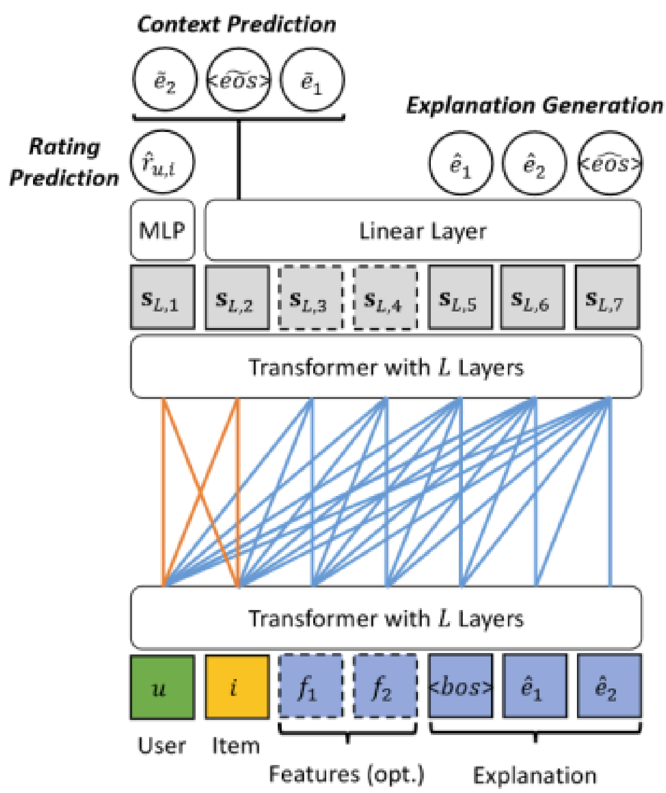


Figure 9: Transformer Based Model

4.4 GPT2 based model

In this section, we present the details of our GPT-2 based model, which is derived from PEPLER[4].As shown in Fig 10, first, we explain our choice of GPT-2 for this task. Next, we demonstrate how the model handles the explanation generation task as a prompt learning task and the recommendation task as regularization. Finally, we identify several drawbacks of the model that we aim to address in our improved version.

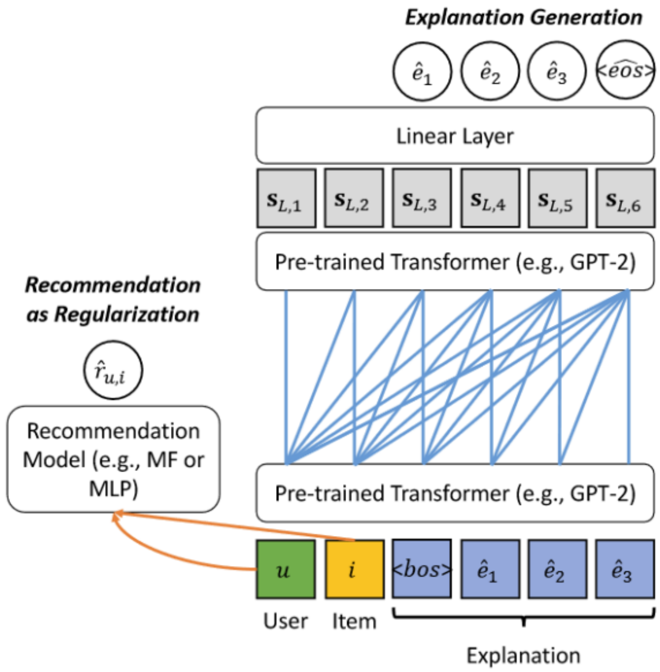


Figure 10: GPT2 Based Model

4.5 Improved GPT2 based model

As shown in Fig 11, inspired by the previous two models, we aim to address their limitations and propose an enhanced version. In this section, we introduce two strategies to improve our model: extended prompts and enhanced multitask learning.

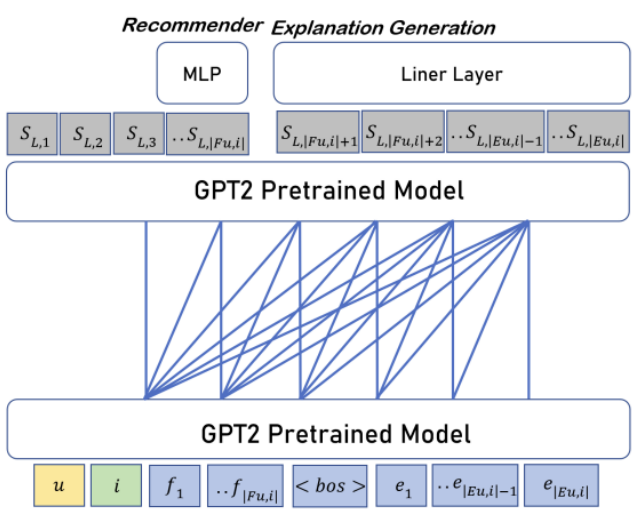


Figure 11: Improved GPT2 Based Model

Extended Prompts: In the previous GPT-2 based model, the prompt was formulated as the vector representation of (user, item). We have now extended this to (user, item, feature). In the extended prompt, the vector representations of the user and item serve the purpose of personalization, aiming to make the generated explanation reflect both the user's interests and the item's attributes. Meanwhile, features can guide the model to discuss certain topics. For instance, a conversational recommender system might explain a recommendation's unique aspects to the user to make the recommendation more reasonable. Adding features to the model also helps the recommender predict ratings more precisely.Thus, the input sequence of our model can be represented as $S =< u, i, f_1, \cdots, f_{|F_{u,i}|}, e_1, \cdots, e_{|E_{u,i}|} >$, where u and i denote the continuous prompt of the user and item. $< f_1, f_2, \cdots, f_{|F_{u,i}|} >$ are the features, and $< e_1, e_2, \cdots, e_{|E_{u,i}|} >$ are the explanation's word sequence. After performing the addition for token representation and positional representation, we obtain $S_0 =< s_{0,1}, s_{0,2}, \cdots, s_{0,|S|} >$ for the input of GPT-2.Thus, it is essential to assess the diversity of the generated explanations. To achieve this, we utilize the Unique Sentence Ratio (USR) metric to compute the diversity of the generated sentences[6]. For the evaluation metrics, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), lower values indicate better performance. Conversely, higher values are preferable for the remaining metrics.

Improved Multitask Learning: To bridge the gap between the two tasks and enhance recommendation performance, we propose a redesigned approach for rating prediction and explanation generation. After obtaining the sequence's final representation $S_L =< s_{L,1}, s_{L,2}, \cdots, s_{L,|S|} >$ from GPT-2, we restructure these tasks to improve the overall system effectiveness.Unlike the previous two models, our approach utilizes the representation $S_{L,2+|E_{u,i}|}$ located at the end of the prompt as the input for a multi-layer perceptron (MLP). This allows us to obtain the estimated rating $\hat{r}_{u,i}$. Similar to the previous model, we apply a linear layer to the final representation of each token to obtain the vocabulary-sized probability distribution. We then adopt the Negative Log-Likelihood (NLL) as the loss function for the explanation task. This approach ensures that the model effectively learns to map each token representation to its corresponding probability, thereby enhancing the accuracy and reliability of the generated explanations.

$$\mathcal{L}_e = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} -log\, c^{e_t}_{2+|F_{u,i}|+t}$$

4.6 Result and Analysis

Table 4 presents the performance comparison of different recommendation methods. Firstly, it is evident that deep learning methods generally outperform traditional collaborative filtering. This superiority can be attributed to the non-linear representation of user preferences offered by deep learning methods, which enables the discovery of unexpected behaviors. Furthermore, deep learning methods effectively extract features from various auxiliary information sources. In our model, we incorporate comprehensive features such as movie reviews, directors, actors, tags, genres, regions, and release years, thereby capturing the intricate characteristics of both users and items to achieve personalized recommendations.Secondly, the GPT-2 based model demonstrates inferior performance compared to other deep learning models. As discussed above, this model relies solely on user ID and item ID information for prediction, resulting in less accurate recommendations. To enhance recommendation performance, our improved GPT-2 based model integrates information extracted from both users and items. This enhancement leads to the model achieving the best performance on the MAE metric, with its RMSE only 0.01 higher than that of the transformer-based model, indicating the success of our improvements in the recommendation task.

Table 4: **Result**

Alg	RMSE	MAE
UCF	1.0066	0.78
ICF	0.9193	0.7427
Transformer	**0.8472**	0.7208
GPT2	1.0322	0.8567
GPT2_Improved	0.8584	**0.6818**

5 Conclusion

The visualization of the social network of the Douban website revealed distinct characteristics compared to other traditional social media platforms. These included a prevalence of discrete nodes and low graph density, alongside typical features of social networks such as short average path length and Zipf-distributed node degrees, indicative of a strong Matthew effect. Our analysis suggests that users primarily utilize Douban for accessing movie information rather than engaging in social interactions, resulting in fewer random follow relationships. Instead, users tend to focus on a select few Douban celebrities for enhanced movie recommendations.

Collaborative filtering faces limitations due to data sparsity, but enhancements such as UCFP with knowledge graph information and adaptive weight assignment significantly improve precision, leading to more personalized recommendations. Integrating social network information into recommendation systems, as demonstrated by ICFP, yields more interpretable recommendations.

To further refine recommendations, we propose a novel NLP pre-processing method to extract valuable review sentences and apply three deep learning models: Transformer-based, GPT2-based, and an improved GPT2-based model. Our experiments show that the improved model outperforms others in both recommendation and explanation tasks.

Future work should focus on incorporating user-specific information into explanations. For example, incorporating user history into explanation templates, such as "you have watched... , and this movie is... , so we recommend it to you," could enhance user engagement and satisfaction.

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292, 2014.

[3] Lei Li, Yongfeng Zhang, and Li Chen. Personalized transformer for explainable recommendation. *arXiv preprint arXiv:2105.11601*, 2021.

[4] Lei Li, Yongfeng Zhang, and Li Chen. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4):1–26, 2023.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[6] Shikib Mehri and Maxine Eskenazi. Usr: An unsupervised and reference free evaluation metric for dialog generation. *arXiv preprint arXiv:2005.00456*, 2020.