Personalized Recommendation System Based on Douban movie

Junming Zhao
NLP data process
deep learning
20307110324

Jianzhi Shen
Data crawling and wash
Collaborative Filtering
20307130030

Yuesen Liao
Data wash,Visualization
Vedio Clip,Report write
20307110128

Abstract

This project conducts social network analysis and personalized movie recommendation on the Douban platform.We visualize two different social networks of the Douban website, and find some network characteristics of the Douban community. Collaborative filtering is implemented and improved. We also implement three deep learning models including the Transformer based model, GPT2 based model and the improved GPT2 based model. We find that our improved model performs best both on the recommendation task and explanation task.

Keywords: Douban movie, social network, recommendation, Collaborative Filtering, GPT2

1 Introduction

1.1 Background

With the popularization of the Internet, people's entertainment methods are becoming more and more abundant, and movies have become an indispensable part of people's daily life. But in the face of a large number of movies, how do video sites recommend suitable movies to users in a personalized way, and how do users find their favorite movies? And more importantly, how to provide these movies with a compelling reason? Based on the above problems, our group implements a personalized recommendation system through the Douban movie dataset.

Douban Movies, as the most authoritative and high-quality movie rating website in China, has collected a large number of movies and related ratings and comments, which has greatly helped the construction of our recommendation system. Formed social networks can also help recommender systems.

1.2 Datasets Overview

1.2.1 Datasets of movies, users and reviews

The main data is from http://moviedata.csuldw.com/ ,an open data source collected in early September 2019.It has a total of 140,502 movie records, 639,125 user records and 4,428,475 comment records.Compared with the data set on elearning , this data set contains more information. Therefore, We can add more feature into the recommendation system to improve the recommendation effect.

The original data has a huge amount of records and also contains a lot of noise. We cleaned it and selected movies with the first 10,000 comments as well as 1,000 users randomly.The comments were limited to these movies and users to obtain 82,894 comment records, which constituted the data set reviews_selected.csv we actually used

1.2.2 Datasets of social network

Regarding the Douban user-fan relationship social network, we have constructed two datasets, one of which is from an open-source dataset on github. [1]The data is composed of users and their follow relationships randomly crawled on Douban website, with a total of 1,000 users and 589 follow relationships.

Regarding the second data set,we select some users in the above data set as seed nodes, then use our crawler program to obtain user's follow relationships. The network has a total of 1,000 users and 15,432 follow relationships, which is much denser than the former one.

2 Social Network Analysis and Visualization

2.1 Movie Dataset

In the data set consisting of the top 10,000 most popular movies, we define popularity as the number of times a comment about the movie appears in the comment subset, and the

[1]https://github.com/jiancgao/Douban-Movie-Recommendation/blob/master/Data/ Douban / douban.sql

visualization result is shown in Figure 1.Then we visualize the word cloud of movie-related tags, and the result is shown in Figure 2.

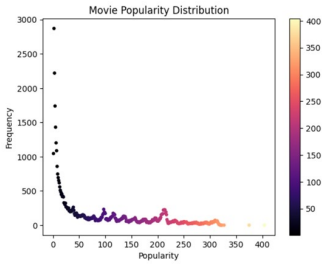


Fig. 1: The observation result shows that the popularity of movies satisfies the Zipf distribution, fewer movies have higher popularity.

Fig. 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:

Parameters	number
nodes	1000
edges	589
density	0.016%
diameter	11
average path length	3.957

Table 1: specific parameters

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

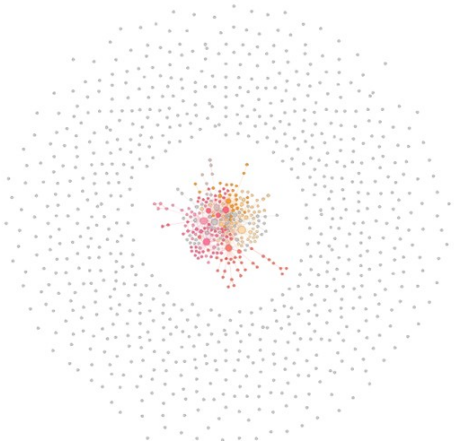


Fig. 3: Random Selected Social Network

Observations found that:

1. There are many discrete nodes, which shows that Douban is not a traditional social networking site like QQ, Weibo, etc.

2. The nodes' size reflects their degree. Only a few nodes are larger in size.

3. The color of nodes represents the community it belongs to, and the community detection is done by the Fast unfolding algorithm . See below for details .

2.2.1 Central group

After removing the discrete nodes, we use two layouts, Force Atlas and Force Directed Layout to display the central community in the network, which are in Figure 4.The former nodes are more concentrated and can intuitively reflect the division of different communities in the network, while the latter can better reflect the distance between nodes.

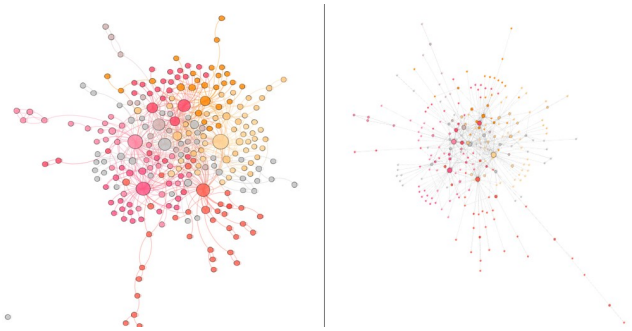


Fig. 4: central group

2.2.2 Fast unfolding algorithm

The Fast Unfolding algorithm(which is also called Louvain algorithm) is an iterative algorithm whose main goal is to continuously divide the community so that the modularity of the entire network after division continues to increase. [Blondel et al.2008],

Modularity is defined as below:

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

The algorithm process is mainly divided into two stages:

The first stage is called Modularity Optimization, which mainly divides each node into the community where its adjacent nodes are located, so that the value of modularity will continue to increase;

The second stage is called Community Aggregation, which mainly aggregates the communities divided in the first stage into one point and reconstructs the network. Repeat the above process until the network's structure no longer changes.

2.3 Seed-Nodes-Started Social Network

2.3.1 Overview

We use the above algorithm to visualize the second social network with Gephi and the results are shown in Figure 5. It divides the network into 7 communities with a modularity of 0.348.

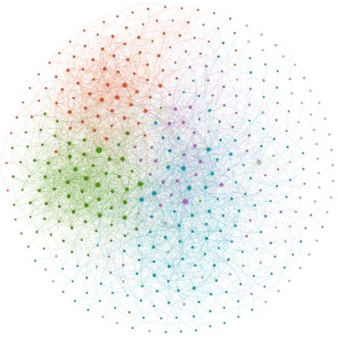


Fig. 5: Gephi

The specific parameters of the social network are as follows:

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

Table 2: specific parameters

2.3.2 Parameters Visualization

Next, we visualize some parameters of the network, and the results are displayed in Figure6,7,8.

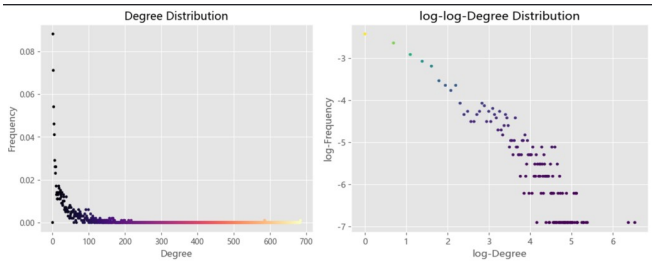


Fig. 6: Degree distribution

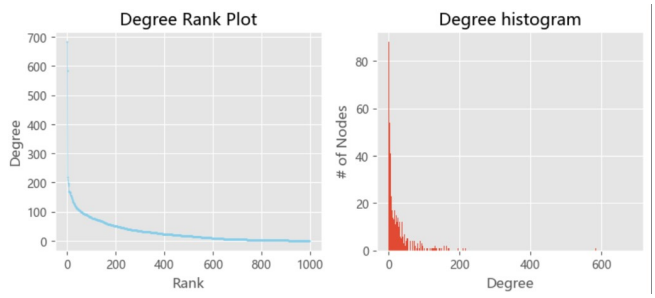


Fig. 7: Degree distribution

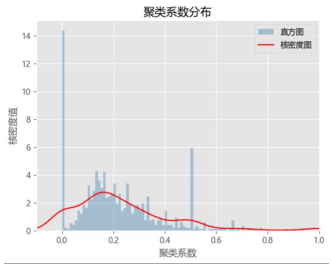


Fig. 8: Clustering Coefficient

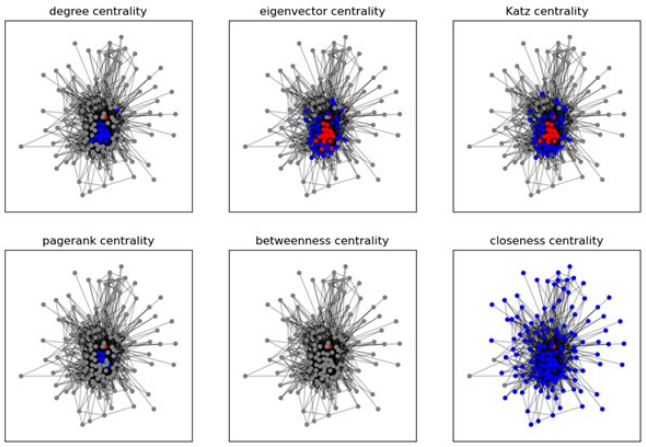
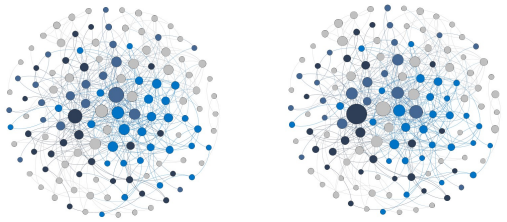


Fig. 9: 6 types of centrality measurements



(a) degree centrality (b) PageRank centrality

Name	degree
beard	37
sickrain	33
liudandan	26
mylife	26
woniu3	22

(c)

Name	PageRank
sickrain	0.05364
liudandan	0.03316
woniu3	0.02924
beard	0.02525
mylife	0.02453

(d)

Fig. 10: 2 types of centrality

We can see that the degree and clustering coefficient also follows Zipf distribution.

2.3.3 Centrality Measurement

We used networkx to measure six types of centrality of the network. The visualization results are shown in Figure9, where the color represent the size of its centrality, red represents its centrality greater than 50% of the maximum value, blue represents greater than 30%, and gray represents less than 30%.

We can see that different measurements lead to quite different results and the majority of nodes are gray in the first 5 graphs.

Finally, we randomly selected nodes to measure degree and pagerank centrality with Gephi, and intercept the data of the top 5 users. The results are shown in Figure 10.

The results of 2 measurement algorithms are quite the same,including their nodes' distribution and their top 5 users.

3 Collaborative Filtering

3.1 Framework

The principle of collaborative filtering lies in homogeneity. Users with similar purchasing behavior tend to be equally interested in an item, and a user tend to choose items which have similar sales record.

Therefore, given M users, N items and the sales or rating record—an $M \times N$ matrix R—among them, from which the purchasing behavior of user u can be easily derived as the u-th row and the sales record of item i, the i-th column, the User-based Collaborative Filtering (UCF, for short) calculates the similarity between every two users u_a and u_b with their purchasing behavior, and predicts the behavior of u on i as a weighted average of the other users' behavior on j, where the more similar i and another user o are, the more the weight of o's behavior is. The Item-based Collaborative Filtering (ICF, for short) calculates the similarity between every two items i_a and i_b with their sales record and predicts the behavior of u on i based on other items u has purchased.

Since the rating record from the test data set is discrete among $1, 2, 3, 4, 5$, we choose Pearson correlation coefficient as the measure of similarity, which can discriminate like and dislike while cosine similarity cannot. The case of UCF is shown in Equation 1, and ICF is similar.

$$sim(u_a, u_b) = \frac{\sum_k (r_{ak} - \bar{r}_a)(r_{bk} - \bar{r}_b)}{\sqrt{\sum_k (r_{ak} - \bar{r}_a)^2 \sum_k (r_{bk} - \bar{r}_b)^2}} \quad (1)$$

The computational complexity of UCF is $M^2 \times N$ and that of ICF is $M \times N^2$. Therefore, on cinecism and social platforms like Douban, where users far outnumber items and users' preference is relatively personal and stable, UCF performs better, while ICF is a better choice on news platform where preference is more social and item update is more frequent. However, UCF can be embedded with users' social information and thus provides recommendation with higher explainability, which will be elaborated in the next section. Therefore, we implemented both ICF and UCF on our data set.

There are some common problems with CF, such as the impact of sparsity on accuracy, the restrict of cold boot and

lack of explainability. The restrict of cold boot is also reflected in the evaluation—if the train set misses all purchasing behavior of a user, then his ratings will be uniformly predicted as 3; if the train set misses all sales record of an item, then its rating from each user will be predicted as the average rating of the user, and every predicted rating will contribute to the final RMSE/MAE, if it is in the test set.

In order to fix these problems to some extent, we introduced some improvements. ICF can be embedded with information from social network and UCF can be embedded with information from knowledge graph. The two improved algorithms, namely ICFP and UCFP, will be elaborated in the next two sections.

3.2 ICFP

We have tested the effect of social network bounded ICF recommendation—only consider the similar users within the neighbors, but due to the sparsity of both rating record and social network, the usable reference is further reduced, so the result is unsatisfactory.

The next trial is introducing node similarity in the social network as another dimension. The user similarity can thus be set as a weighted average of behavioral similarity and node similarity. However, the result is still far from ideal. One possible explanation is that since Douban is simultaneously a social platform, many connections may not be based on similar taste in film, but merely experience sharing, so considering all the connections will make the similarity matrix kind of noisy.

However, on such platforms users with high centrality may be senior film critics and thus provide more latent information. A user follows a senior film critic because he is interested in most films reviewed by the critic, and finds the critic's reviews objective (which is, in fact, subjective). Therefore, we can find the node set H with high-centrality elements, measure the similarity between two users u_a and u_b by the coincidence of followed high-centrality nodes with Jaccard similarity as shown in Equation 2. Moreover, when predicting a user's rating, the weight of a similar user who is also a high-centrality neighbor can be further raised.

$$sim_H(u_a, u_b) = \frac{|N(a) \cap N(b) \cap H|}{|N(a) \cup N(b) \cap H|} \tag{2}$$

3.3 UCFP

In UCFP we embedded UCF with knowledge graph, and focused on three main properties, directors, actors and genres of a movie. The similarity over each property is calculated as Equation 3 [Xiao et al.2014]

$$sim_p(i_a, i_b) = \frac{2|S_p(a) \cap S_p(b)|}{|S_p(a)| + |S_p(b)|} \tag{3}$$

Then how to combine the three similarity measure remains to be discussed. Our approach is, adaptively assigning the weights based on the users' image, i.e. different users, different weights assigned, since some users may care more about the director, some may be a huge fan for certain actors, and others may have a craze for certain genres.

To construct a user's image, we select a range of entities and calculate his affection for each entity t, which is defined as Equation 4, where $\bar{r}_t$ is the average rating he gives to items related to t, and $freq_t$ is the proportion of items related to t in his purchasing history. If a user highly rates a single item related to t by chance, this measure can prevent mistaking him as a lover of t.

$$A_t = \bar{r}_t + 5\sqrt{freq_t} \tag{4}$$

An the variance of A_t within a property, as shown in Equation 5 measures how much he cares about the property. Therefore, V_p can be the weight of sim_p when predicting for this user.

$$V_p = Var_{t \in p}(A_t) \tag{5}$$

3.4 Result

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

Table 3

As is shown in Table 3, due to sparsity, the performance of UCF is unsatisfactory, and UCFP did not improve UCF in precision very much. Since the social network is sparse, the information it provides is limited. Fortunately, the recommend result (the top 5 items among the predicted) becomes more explainable. ICF does better than UCF, which indicates that on this platform preference is personal rather than social. ICFP further improves ICF, because it digs deeper the similarity between items and simultaneously consider the difference among users.

4 Deep Learning Method

In this part, we will introduce three deep learning models(transformer based, GPT2 based, improved GPT2 based) for Explainable Recommendation task.The first 2 models are respectively referred from PETER(Li and Zhang, 2021) and PEPLER(Li and Zhang, 2022), while the third model is posed by us, showing the best performance on this task. To make the models more efficient on douban data set. We also introduce a NLP preprocess method to extract valuable information from user's reviews.

4.1 Problem Formulation

Given user u and item i our goal is to estimate a rating $\hat{r}_{u,i}$ that predicts u ' s preference towards i. Meanwhile, our model can also generate a natural language sentence $\hat{E}_{u,i}$ for a pair of user u and item i to justify why i is recommended to u. To make the recommendation and explanation more precise, we involve the features $F_{u,i}$ including the movie's tag, genres, director, actors, region and release date. In the training stage, we should input the explanation $E_{u,i}$, while in the generating stage we replace it with <eos> to let the model generate explanation.

4.2 NLP Pre Process

Since there is lots of noise in the raw data. To obtain the suitable explanation $E_{u,i}$ from each review. we will extract top 3 most valuable sentences as the explanation. We think that the sentence which holds the following two characters is better: 1.it can indicate the user's preference; 2.it is highly related to the basic information of the movie. For instance, "Cameron's special effects are attractive" might be a suitable reason to recommend Avatar. So we derive two scores below.

similarity score $score\_sim$ evaluates the similarity between the review sentence s and the movie introduction document D. Movie introduction written by movie producer concludes the basic information of the movie. The more similar the sentence is to the introduction, the more likely it will contain basic information about the movie. We compute $score\_sim$ by BM25 algorithm:

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

where $q_1, q_2, ..., q_n$ are the words contained in the sentence s, $f(q_i, D)$ is the number of times that q_i occurs in the document D, $\|D\|$ is the length of the document D in words, and avgdl is the average document length in the text collection from which documents are drawn. k_1 and b are hyper parameters.

sentiment score $score\_sentiment$ evaluates the how well a sentence fits the user's sentiment. We use the snownlp sentiment rating, which is a trained bayesian classifier. Enter a sentence s, it will scan all the emotional keywords, and use the Bayesian classifier to give a score $sentiment(s)$ in the 0-1 interval. higher $sentiment(s)$ indicates more positive emotion. Then we compute the $score\_sentiment$ as follow.

$$score\_sentiment(s) = -\|sentiment(s) * 5 - rating\| \quad (6)$$

Where $rating$ is the user's rating in this review, ranging from 1 to 5 that reflects user's preference towards a movie. The higher the sentiment score is ,the more likely the sentence fits the user's sentiment. For each sentence s in a review, we combine the two scores to get the total score, and extract the top 3 sentence to get 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 [Vaswani et al.2017] based model which is referred from PETER [Li et al.2021]. First, we show how to encode different types of tokens in a sequence. Then, we introduce model's frame and revised attention masking matrix. At last, we formulate the three tasks, i.e., explanation generation, context prediction and recommendation, and integrate them into a multi-task learning framework.

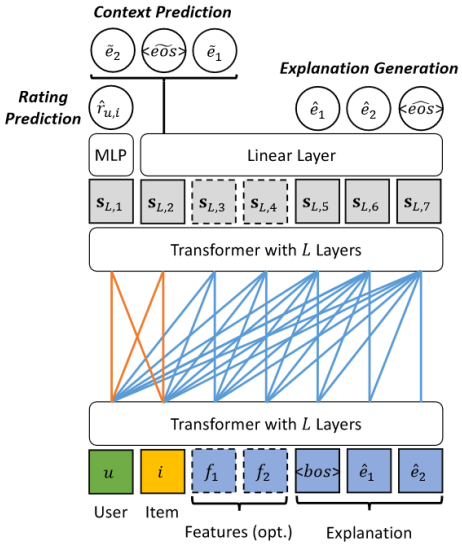


Fig. 11: The transformer based model

4.3.1 Input Representation

As shown in Fig. 11, 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 $S = \left[u, i, f_1 ... f_{|F_{u,i}|}, e_1 ... e_{|E_{u,i}|}\right]$, where $\left[f_1, f_2 ... f_{|F_{u,i}|}\right]$ are the features and $\left[e_1, e_2 ... e_{|E_{u,i}|}\right]$ are the explanation's word sequence. $\left|F_{u,i}\right|$ denotes the number of features and $\left|E_{u,i}\right|$ is the number of words in the explanation. Clearly there are three types of tokens in the sequence S, i.e., users, items, and words (including features), for which we prepare three sets of randomly initialized token embeddings **U**, **I** and **V** respectively, besides the positional embeddings **P** that encode the position of each token in the sequence.After performing embedding lookup and positional encoding, we can obtain the input sequence $S_0 = \left[s_{0,1}, s_{0,2} ... s_{0,|S|}\right]$, where $|S|$ is the length of the sequence.

4.3.2 Model Framework

As is shown in Fig. 11, our model contains 4 parts: embedding layer, 2 transformer encoders, MLP for recommendation task, Linear layer for explanation generation task. Transformer encoder consists of L identical layers, each of which is composed of two sub-layers: multi-head self-attention and positionwise feed-forward network. The l-th layer encodes the previous layer's output S_{l-1} into S_l. In the multi-head self-attention sublayer, the h-th head attention $A_{l,h}$ is computed as follows:

$$A_{l,h} = softmax(\frac{Q_{l,h}K_{l,h}^T}{\sqrt{d}} + M)V_{l,h}$$

$$Q_{l,h} = S_{l-1}W_{l,h}^Q \ , \ K_{l,h} = S_{l-1}W_{l,h}^K$$

$$V_{l,h} = S_{l-1}W_{l,h}^V$$

$$M = \begin{cases} 0 & allow\ to\ attend \\ -\infty & not\ allow\ to\ attend \end{cases}$$

Where $W_{l,h}^Q, W_{l,h}^K, W_{l,h}^V \in \mathbb{R}^{|S| \times d}$ are projection matrices, d denotes the dimension of embeddings, and $M \in \mathbb{R}^{|S| \times |S|}$ is the attention masking matrix.

Since we want the model to generate explanation, we still need to use attention mask in our model. Moreover, we will let the first two tokens u and i can attend to each other, because both context prediction and recommendation tasks need them. The attention mask is shown in Fig. 12.

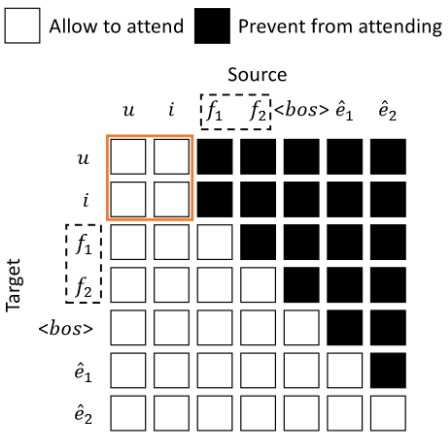


Fig. 12: The masking strategy

4.3.3 Multitask learning

After obtaining the sequence's final representation $S_L = \left[s_{L,1}, s_{L,2} ... s_{L,|S|}\right]$ from Transformer. We formulate the three tasks as follows:

Rating Prediction: As both u and i in the sequence can attend to each other, their final representations capture the interaction between them. So we use the 1st representation $s_{L,1}$ as the input of multi-layer perceptron (MLP) to obtain the estimated rating $\hat{r}_{u,i}$:

$$\hat{r}_{u,i} = w^r \sigma(W^r s_{L,1} + \beta^r) + b^r$$

Where $W^r \in \mathbb{R}^{d \times d}$ $\beta^r \in \mathbb{R}^d$ $w^r \in \mathbb{R}^{1 \times d}$ and $b^r \in \mathbb{R}$ are weight parameters, $\sigma(.)$ is the sigmoid function. we use Mean Square

Error (MSE) as the loss function:

$$\mathcal{L}_r = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} (r_{u,i} - \hat{r}_{u,i})^2 \tag{7}$$

where $r_{u,i}$ is the ground-truth rating.

Explanation Generation: In this tasks, we apply a linear layer to the final representation of each token to map it into a $|\mathcal{V}|$ -sized vector. As an example, after passing through this layer, $s_{L,t}$ becomes c_t:

$$c_t = softmax(W^v s_{L,t} + b^v) \tag{8}$$

Where $W_v \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $b_v \in \mathbb{R}^{|\mathcal{V}|}$are weight parameters. The vector c_t represents the probability distribution over the vocabulary $\mathcal{V}$, from which a word e with probability c_t^e can be sampled.

We adopt the Negative Log-Likelihood (NLL) as the explanation task' s loss function, and compute the mean of user item pairs in the training set:

$$\mathcal{L}_e = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} -log\, c^{e_t}_{2+|F_{u,i}|+t} \tag{9}$$

Where $\mathcal{T}$ denotes the training set. The probability $c_t^{e_t}$ is offset by $2 + |F_{u,i}|$ positions because the explanation is placed at the end of the sequence.

At the testing stage, along with u, i, and $F_{u,i}$ (if available), we feed the model a special begin-ofsequence token <bos>. From its resulting probability distribution $c_{<bos>}$, the model can predict a word. We opt for greedy decoding that samples the word with the largest probability.Then we can concatenate this predicted word at the end of the sequence to form a new input sequence for generating another word. We do this repeatedly until the model produces a special end-of-sequence token <eos>, or the generated explanation $\hat{E}_{u,i}$ reaches a pre-defined length.

Context Prediction: we use the 1st representation $s_{L,1}$ as the input of linear layer to estimate the explanation, Again, we adopt NLL as the loss function:

$$\mathcal{L}_c = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} -log\, c_2^{e_t} \tag{10}$$

The only difference from Eq. 9 is that all predicted words are from the 2nd position. The purpose of this task is to increase the importance of the user u and item i in the attention. Since we want the generated explanation to be more related to user and item, not just talking about the features. We design this task to map the IDs onto the words in the explanation, so as to build a connection between them.

Multi-task Learning: At last, we integrate the three tasks into a multi-task learning framework whose objective function is defined as:

$$\mathcal{J} = \min_{\Theta} (\lambda_e \mathcal{L}_e + \lambda_c \mathcal{L}_c + \lambda_r \mathcal{L}_r) \tag{11}$$

where Θ denotes all the trainable parameters in the model, and λ_e, λ_c and λ_r are regularization weights that balance the learning of different tasks. In this way, the model can be trained efficiently in an end-to-end manner.

4.4 GPT2 based model

In this section, we present the details of our GPT2 based model which is referred from PEPLER [Li et al.2022]. First, we explain why we choose GPT2, then we will show how the model handle the explanation generation task as a prompt learning task an recommendation task as Regularization. At last, we point out several drawbacks of the model which we try to cover in our improved model

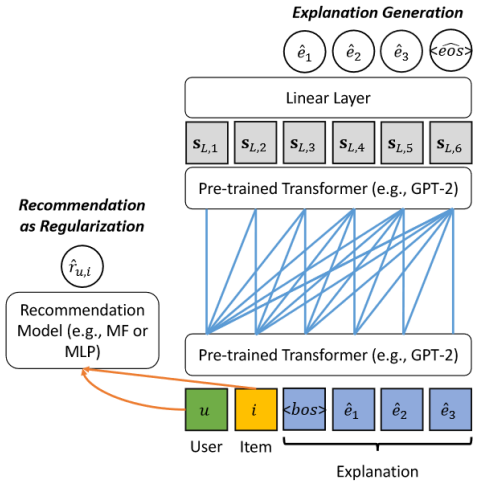


Fig. 13: The GPT2 based model

4.4.1 Why GPT2

In the former discussion, we know that the goal of explanation generation task here is to generate a textual explanation $\hat{E}_{u,i}$ given user u and item i. In the generating stage, we make the next-token prediction based on preceding tokens in an auto-regressive method. For instance, at time t, we have

obtained estimated explanation token $[e_1, e_2...e_{t-1}]$, we then concatenate user, item and these tokens as the input of t times: $[u, i, e_1, e_2...e_{t-1}]$, we then generate e_r as follows:

$$e_t = argmax\, log\, P(e|u, i, e_1...e_{t-1}) \tag{12}$$

GPT2, which is based on the Transformer decoder, uses the Left-to-Right Masking method. Specifically, in these models, the lower triangular part of attention mask matrix **M** is set to 0 and the remaining part ∞, so as to allow each token to attend to past tokens (including itself), but prevent it from attending to future tokens. It focus on predict tokens based on preceding tokens. Thus GPT2 performes well in the textual explanation generation task.

4.4.2 Explanation Generation as Prompt Learning

prompt learning method aims at using prompt information(e.g. natural language text) to guide pre-trained language models to handle specific task. Here, we regard information of user and item as a prompt that will guide the GPT2 to generate explanation based on user and item.

Since the word tokens on which GPT2 is trained are inherently in a different semantic space as ID tokens, we map user and item ID to vector representations. In this way, ID vectors could be directly used as continuous prompts to generate recommendation explanations. Specifically, we prepare two sets of token embeddings:$U \in \mathbb{R}^{|\mathcal{U}|\times d}$ and $I \in \mathbb{R}^{|\mathcal{I}|\times d}$ where $\mathcal{U}$ and $\mathcal{I}$ respectively represent the number of users and items in a dataset. Then, a user u's vector representation can be retrieved via:

$$\mathbf{u} = U^T g(u) \tag{13}$$

where $g(u) \in \{0, 1\}^{|\mathcal{U}|}$ denotes a one-hot vector, whose non-zero element corresponds to the position that user u' s vector locates in U. In a similar way, we can obtain **i** from I for item i.Then, the sequence' s token representation can be denoted as $\left[\mathbf{u}, \mathbf{i}, e_1...e_{|E_{u,i}|}\right]$

The follow-up steps are similar to what we do in section 4.3.1:performing addition for token representation and positional representation to obtain $S_0 = \left[s_{0,1}, s_{0,2}...s_{0,|S|}\right]$. passing S_0 through pre-trained GPT2 for producing $S_L = \left[s_{L,1}, s_{L,2}...s_{L,|S|}\right]$, applying a linear layer with softmax function to each token' s final representation $s_{L,t}$ for next-word prediction, and employing NLL loss function on the word probability distribution c_t

$$\mathcal{L}_c = \frac{1}{|\mathcal{T}|}\sum_{u,i\in\mathcal{T}}\frac{1}{|E_{u,i}|}\sum_{t=1}^{|E_{u,i}|} -log\, c_2^{e_t} \tag{14}$$

where $c_2^{e_t}$ is offset by 2 positions (i.e., user ID and item ID), which is slightly different multiple positions of features in Eq. 10.

4.4.3 Recommendation as Regularization

To do the recommendation task, we use the output of user and item embedding **u** and **i** as the input of MLP. After obtaining the estimated rating $\hat{r}_{u,i}$, we use the mse as loss function:

$$\mathcal{L}_r = \frac{1}{|\mathcal{T}|}\sum_{u,i\in\mathcal{T}}(r_{u,i} - \hat{r}_{u,i})^2 \tag{15}$$

where $r_{u,i}$ is the ground-truth rating that user u assigned to item i. The recommendation task in this just serve as regularization to help train the embedding layer of user and item, the model's training goal is to minimize the total loss:

$$\mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P,\Theta_R\}}(\mathcal{L}_c + \lambda_r\mathcal{L}_r) \tag{16}$$

where the model parameters Θ consist of pre-trained language model parameters Θ_{LM}, continuous prompt parameters Θ_P and recommendation model parameters Θ_R.

4.4.4 Drawbacks

For the recommendation task, the model only use the information in user id and item id. Without more features about user and item being added, the prediction of ratings may not be accurate enough. We will show this later in the experiment part.

For the interaction between two tasks, we think they help each other very little. Since the input of the MLP is just the embedding of user and item, explanation task has little effect on recommendation task. Moreover, among the parameters related to explanation task, recommendation only help the learning of parameters Θ_P in embedding layer, more parameters Θ_{LM} in GPT2 will not be affected.

4.5 Improved GPT2 Based Model

Inspired by the previous two models, we try to cover the drawbacks and propose our improved model. In this section, we will introduce two strategies to improve our model, i.e. extended prompts and improved multitask learning.

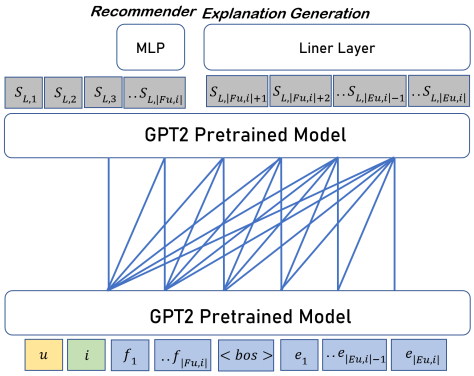


Fig. 14: The improved GPT2 based model

4.5.1 Extended Prompts

In the previous GPT2 based model, the prompt is formulated as the vector representation of (user, item). Now we extended it to (user, item, feature). In the extended prompt, The vector representation of user and the item serve for the purpose of personalization, i.e., aiming to make the generated explanation reflect both the user's interests and the item's attributes. While features can guide the model to talk about certain topics. For instance, a conversational recommender system may explain a recommendation's specialty to the user with the goal of making the recommendation more reasonable. Adding features to model can also help recommender to predict rating more precisely.

Thus, The input sequence of our model can be represented as $S = \left[\mathbf{u}, \mathbf{i}, f_1 ... f_{|F_{u,i}|}, e_1 ... e_{|E_{u,i}|}\right]$, where $\mathbf{u}$ and $\mathbf{i}$ denotes the continous prompt of user and item. $\left[f_1, f_2 ... f_{|F_{u,i}|}\right]$ are the features and $\left[e_1, e_2 ... e_{|E_{u,i}|}\right]$ are the explanation's word sequence. After performing addition for token representation and positional representation, we obtain $S_0 = \left[s_{0,1}, s_{0,2} ... s_{0,|S|}\right]$ for the input of GPT2.

4.5.2 Improved Multitask Learning

To overcome the gap between two task and improve the recommendation performance, after obtaining the sequence's final representation $S_L = \left[s_{L,1}, s_{L,2} ... s_{L,|S|}\right]$ from GPT2, we redesign the rating prediction and explanation generation task as follows:

Rating Prediction: Unlike the previous two models, we use the representation $s_{L,2+|E_{u,i}|}$ located at the end of prompt as the input of multi-layer perceptron (MLP) to obtain the estimated rating $\hat{r}_{u,i}$.

$$\begin{cases} a_0 = \sigma(W_0 s_{L,2+|E_{u,i}|} + b_0) \\ a_1 = \sigma(W_1 a_0 + b_1) \\ \cdots \\ a_N = \sigma(W_N a_{N-1} + b_N) \end{cases} \tag{17}$$

$$\hat{r}_{u,i} = w^T a_N + \beta \tag{18}$$

Where $W_0 \in \mathbb{R}^{d \times d_h}$, $W_* \in \mathbb{R}^{d \times d}$, $b_* \in \mathbb{R}^d$, $w \in \mathbb{R}^d$, $\beta \in \mathbb{R}$ are weight parameters. we use Mean Square Error (MSE) as the loss function:

$$\mathcal{L}_r = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} (r_{u,i} - \hat{r}_{u,i})^2 \tag{19}$$

where $r_{u,i}$ is the ground-truth rating.Since the token located at the end of prompt is allowed to attend to the whole prompt, the GPT2 output $s_{L,2+|E_{u,i}|}$ contains the extracted information about user, item and feature. Using it as the input of MLP will improve recommendation accuracy.

Explanation Generation: Similar as the previous model, we apply a linear layer to the final representation of each token to get the vocabulary-sized probability distribution c_t. Then We adopt the Negative Log-Likelihood (NLL) as the explanation task's loss function,:

$$\mathcal{L}_e = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} -log\, c^{e_t}_{2+|F_{u,i}|+t} \tag{20}$$

Where the probability $c_t^{e_t}$ is offset by $2 + |F_{u,i}|$ positions because we have added fetures into our prompt.

Multi-task Learning: At last, we integrate the two tasks into a multi-task learning framework whose objective function is defined as:

$$\mathcal{J} = \min_{\Theta = \{\Theta_{LM}, \Theta_P, \Theta_R, \Theta_L\}} (\lambda_e \mathcal{L}_e + \lambda_r \mathcal{L}_r) \tag{21}$$

Where the model parameters Θ consist of pre-trained language model parameters Θ_{LM}, continuous prompt parameters Θ_P, recommendation model parameters Θ_R and linear layer parameters Θ_L. In the training stage, both recommendation task and explanation generation task will help the learning of Θ_{LM} and Θ_P. Since the recommendation task has involved in the fine tuning of GPT2, the two tasks are sharing much more parameters than previous model. The gap between the two tasks will be bridged.

4.6 Evaluation Metrics

To evaluate the recommendation performance, we adopt two commonly used metrics: Root Mean Square Error

(RMSE) and Mean Absolute Error (MAE). As to explanation performance,we measure the generated explanations from two main perspectives: quality and diversity. For the former, we adopt BLEU [Papineni et al.2002] in machine translation and ROUGE lin-2004-rouge in text summarization, and report BLEU-1 and BLEU4, and Precision, Recall and F1 of ROUGE-1 and ROUGE-2. Though being widely used, BLUE and ROUGE are not flawless. For example, it is difficult for them to detect the problem of identical sentences generated by GPT2. These identical sentences might not be used as explanations, because they are less likely to well explain the special property of different recommendations. Thus we need to evaluate the diversity of generated explanation. we adopt USR that computes the Unique Sentence Ratio of generated sentences [Mehri and Eskénazi2020].

For RMSE and MAE, the lower, the better, while it is opposite for the rest of metrics.

4.7 Result and Analysis

4.7.1 Recommendation Performance

Table 4 presents the performance comparison of different recommendation methods. First, we can see that the deep learning methods generally perform better than the traditional collaborative filtering. The main reason is that the deep learning methods provide a non-linear representation of user preferences, which can discover unexpected behaviors. Moreover, the deep learning method can effectively extract features from a variety of auxiliary information. In our model, we add the movie's review, directors and actors, tags, genres, regions and year, comprehensively consider the characteristics of users and items, and realize personalized recommendations.

Secondly, the GPT2 based model performs worse than other deep learning models. Just as mentioned in section 4.4.4, the model only uses the information of user id and item id to make prediction, which will makes the recommendation less accurate. To improve the recommendation performance, our improved GPT2 based model uses the information extracted from both user item and features. It performs best on MAE metrics and the it's RMSE is only 0.01 higher than the transformer based model, indicating that our improvement has succeeded in the recommendation task.

4.7.2 Quantitative Analysis on Explanations

The performance comparison between different models is shown in Table 5. As shown in the table, out of the nine selected metrics, our improved model performed best on seven.

	RMSE	MAE
UCF	1.0066	0.78
ICF	0.9193	0.7427
Transfomer	**0.8472**	0.7208
GPT2	1.0322	0.8567
GPT2_improved	0.8584	**0.6818**

Table 4

To be more precise, on the metrics of text quality (i.e. BLEU and ROUGE), both GPT2 based model and our improved model performs far better than the Transformer based model, showing the advantages of GPT2 in generating text.

For the diversity metrics of explanation, the USR of our improved model is far better than others. This is because we let the recommendation task participate in the fine-tuning of GPT2, letting the two tasks have more connections. Since the recommendation task focuses on predicting user's preference according the feature of user and item, it will guide GPT2 to generate different explanation according to different preference, making the explanation more reasonable. However, when there is gap between two tasks, the GPT2 based model tend to generate similar explanation regardless of whether the user like the movie or not.

4.7.3 Qualitative Case Study on Explanations

To compare the models more specifically, we selected some of the generated explanation here. First, choosing the user e58823802b6c708ba8dd3088daf205b6 and movie id 5401681.0, the generated explanation from the three models is shown in Table 6 . We can see that the explanation from our improved model is more readable. Moreover, it tries to talk about the movies feature, making the explanation more informative and reasonable. To compare the diversity of explanation, we select the same movie(id:1301811.0) and compare the explanation from GPT2 based model and our improved model given to different users(id: 1adbf9b484591a90a274314b4a600faf and 802203bdf66816913de05d37633fb2a2). As is shown in Table 7 , the GPT2 based model generates the same explanation to the two users, regardless of the rating they gave to the movie. While the explanations from our improved model are totally different. higher rating corresponds to more positive explanation, showing that our model can generate more personalized explanation referring to his or her preference.

5 Conclusion

We used various methods to visualize the social network of Douban website, and found that it presents different characteristics from other traditional social media, mainly reflected in the existence of a large number of discrete nodes and low graph density. But at the same time, it also has the general characteristics of social networks, such as the short average path length, the Zipf-ditributed node degrees, etc. which indicates its strong Matthew effect.

We concluded that users often use the website to obtain movie information rather than social activities, so there is less follow relationship between random users. They are more inclined to pay attention to a small number of Douban celebrities rather than ordinary users, as the former can provide them with more and better movie information.

Collaborative filtering's performance is limited due to the sparsity. UCFP with information from knowledge graph and adaptive weight assignment improves the precision significantly, and can make more personalized recommendations. ICFP, embedded with social network, gives more explainable recommendations.

To make the recommendation more precise and reasonable, We design NLP pre-process method to extract the valuable review sentences and implement three deep learning models including the Transformer based model, GPT2 based model and the improved GPT2 based model. We find that our improved model performs best both on the recommendation task and explanation task.

For the future work, since our model can only generate explanation concerning the movie's feature. We may add more information about the user to the explanation in the future. For instance, explanation templates like "you have watched..., and this movie is..., so we recommend it to you." may be more attractive.

References

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

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *ACL*.

Lei Li, Yongfeng Zhang, and Li Chen. 2022. Personalized prompt learning for explainable recommendation. *arXiv preprint arXiv:2202.07371*.

Shikib Mehri and Maxine Eskénazi. 2020. Usr: An unsupervised and reference free evaluation metric for dialog generation. *ArXiv*, abs/2005.00456.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Y. Xiao, R. Xiang, Y. Sun, Q. Gu, and J. Han. 2014. Personalized entity recommendation: a heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*.

explanation generation performance									
Model	BLEU-1	BLEU-4	rouge_1/f	rouge_1/r	rouge_1/p	rouge_2/f	rouge_2/r	rouge_2/p	USR
Transfomer	3.2329	0.1471	5.3321	3.1438	27.7503	0.5964	0.3476	3.3534	0.042
GPT2	21.5379	3.419	**26.5516**	24.0132	**39.0629**	9.6655	8.8315	14.1406	0.0153
GPT2_improved	**22.1854**	**3.9869**	26.4734	**24.6847**	35.4457	**10.0586**	**9.4962**	**15.3566**	**0.5025**

Table 5

generated text comparison	
user id:e58823802b6c708ba8dd3088daf205b6 movie id: 5401681.0	
Transformer	香港香港香港香港香港香港香港香港香港香港香港香港香港香港香港香港香港香港
GPT2	[CLS] 但是剧情还是不错的虽然剧情不够深入但是
GPT2_improved	<bos>[CLS] 李小龙的电影总是能让人感到一种情怀 [SEP]

Table 6

generated text comparison on movie id:1301811.0			
	GPT2	GPT2 improved	rating
user1	<bos>[CLS] 不过这个故事的确是一个不错的故事 [SEP]	<bos>[CLS] 很好的儿童片 [SEP]	4
user2	<bos>[CLS] 不过这个故事的确是一个不错的故事 [SEP]	<bos>[CLS] 但是这样的故事却不是我的菜	3

Table 7