โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ การรวมทรานส์ฟอเมอร์กับเจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับการแนะนำแบบกรองร่วม
Incorporate Transformers with Generative Adversarial Networks for User-based and Item-based Collaborative filtering recommendation

ชื่อนิสิต นางสาวธันย์ชนก ตั้งปอง 593 36323 23

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2562

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

การรวมทรานส์ฟอเมอร์กับเจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับการแนะนำแบบกรองร่วม

นางสาวธันย์ชนก ตั้งปอง 593 36323 23

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Incorporate Transformers with Generative Adversarial Networks for User-based and Item-based Collaborative filtering recommendation

Thunchanok Tangpong 593 36323 23

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science Program in Computer Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2019

หัวข้อโครงงาน การรวมทรานส์ฟอเมอร์กับเจนเนอเรทีฟแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับการแนะนำแบบกรองร่วม

โดย นางสาวธันย์ชนก ตั้งปอง

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการหลัก รองศาสตราจารย์ ดร.ศรันญา มณีโรจน์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิตในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

.... หัวหน้าภาควิชาคณิตศาสตร์
(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) และวิทยาการคอมพิวเตอร

คณะกรรมการสอบโครงงาน

.... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. ศรันญา มณีโรจน์) โครงงานหลัก

.... กรรมการ
(รองศาสตราจารย์ ดร. วิมลรัตน์ งามอร่ามวรางกูร)

.... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. กิติพร พลายมาศ)

นางสาวธันย์ชนก ตั้งปอง: การรวมทรานส์ฟอเมอร์กับเจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับการแนะนำแบบกรองร่วม (Incorporate Transformers with Generative Adversarial Networks for User-based and Item-based Collaborative filtering recommendation)
อ.ที่ปรึกษาโครงงานหลัก: รองศาสตราจารย์ ดร.ศรันญา มณีโรจน์ 55 หน้า.

การสกัดตัวแทนเครือข่ายบรรณานุกรมแบบวิวิธพันธุ์โดยใช้เจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับระบบแนะนำการอ้างอิงส่วนบุคคลเป็นระบบแนะนำชนิดหนึ่งที่ประยุกต์เจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คมาใช้ในงานระบบแนะนำ แต่อย่างไรก็ตามการสกัดตัวแทนเครือข่ายบรรณานุกรมแบบวิวิธพันธุ์โดยใช้เจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับระบบแนะนำการอ้างอิงส่วนบุคคลยังมีข้อจำกัดในเรื่องของการสกัดตัวแทน ซึ่งข้อมูลที่นำมาสกัดใช้แค่ข้อมูลของผู้ใช้เป้าหมายและรายการเป้าหมายเท่านั้น โดยไม่ได้พิจารณาข้อมูลของเพื่อนบ้านรอบข้างของผู้ใช้เป้าหมายและรายการเป้าหมายเลย ซึ่งในระบบแนะนำมองว่าเพื่อนบ้านรอบข้างของผู้ใช้เป้าหมายและรายการเป้าหมายค่อนข้างมีผลต่อลักษณะของผู้ใช้เป้าหมายและรายการเป้าหมาย เพื่อที่จะพิจารณาข้อมูลของเพื่อนบ้านรอบข้างของผู้ใช้เป้าหมายและรายการเป้าหมายด้วย ผู้ทำวิจัยจึงใช้วิธีการแนะนำแบบกรองร่วมชนิดผู้ใช้และการแนะนำแบบกรองร่วมชนิดรายการมาใช้ในขั้นตอนวิธีที่นำเสนอมาด้วย นอกจากนี้ผู้ทำวิจัยใช้ทรานส์ฟอเมอร์มาทำหน้าที่เป็นตัวตรวจสอบแทนดีนอยส์ซิงออโต้เอนโคเดอร์ในเจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์ค เนื่องจากดีนอยส์ซิงออโต้เอนโคเดอร์จะให้น้ำหนักกับทุกรายการที่ผู้ใช้เคยให้คะแนนด้วยน้ำหนักที่เท่ากัน แต่ทรานส์ฟอเมอร์จะให้น้ำหนักกับทุกรายการที่ผู้ใช้เคยให้คะแนนด้วยน้ำหนักที่ควรจะเป็น ดังนั้นขั้นตอนวิธีที่นำเสนอมามีการใช้ทรานส์ฟอเมอร์ในเจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คเพื่อสกัดตัวแทนของผู้ใช้และตัวแทนของรายการจากข้อมูลของเพื่อนบ้านรอบข้างของผู้ใช้เป้าหมายและรายการเป้าหมายเลยโดยใช้วิธีของการแนะนำแบบกรองร่วมชนิดผู้ใช้และการแนะนำแบบกรองร่วมชนิดรายการตามลำดับ ตัวแทนของผู้ใช้และตัวแทนของรายการที่ได้จะถูกนำไปทำนายคะแนนเรตติ้ง ผู้วิจัยได้ใช้ชุดข้อมูลของ movieLen Small Latest เพื่อเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอมากับการสกัดตัวแทนเครือข่ายบรรณานุกรมแบบวิวิธพันธุ์โดยใช้เจนเนอเรทีพแอดเวอร์ซาเรียลเน็ตเวิร์คสำหรับระบบแนะนำการอ้างอิงส่วนบุคคล โดยผลการทดลองพบว่าขั้นตอนวิธีที่ผู้วิจัยนำเสนอ ให้ผลโดยรวมที่ดีกว่าในด้านของความถูกต้อง

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต ธันย์ชนก ตั้งปอง
สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงานหลัก ศรันญา มณีโรจน์
ปีการศึกษา 2562

5933632323: MAJOR COMPUTER SCIENCE

KEYWORDS: GENERATIVE ADVERSARIAL NETWORKS, COLLABORATIVE FILTERING, TRANSFORMER, RECOMMENDER SYSTEM

MS. THUNCHANOK TANGPONG: INCORPORATE TRANSFORMERS WITH GENERATIVE ADVERSARIAL NETWORKS FOR USER-BASED AND ITEM-BASED COLLABORATIVE FILTERING RECOMMENDATION. ADVISOR: ASSOC. PROF. SARANYA MANEEROJ, Ph.D. 55 pp.

GAN Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation (GAN-HBNR) is an existing GAN recommendation approach. However, GAN-HBNR is only focused typically on information from the target user or target item for creating user/item representation to predict the score, while ignoring the opinions of other users or neighbors. In common sense, the user's neighbors and item's neighbors make a significant effect on the user's characteristics and item's characteristics. To focus on the user's neighbors and the item's neighbors, I apply user-based collaborative filtering (user-based CF) and item-based collaborative filtering (item-based CF) to my proposed model. Furthermore, I use a transformer instead of denoising autoencoder (DAE) as the discriminator on GAN. Because DAE weighs every item that the target user has rated with the same attention, while transformer weighs every item that the target user has rated with the different attention. Therefore, my proposed model incorporates transformers in a generative adversarial network-based model to learn user representation and item representation that represents the relation between the target user's preference and his/her neighbors' preference to perform user-based CF, and the relation between the target item's preference and its neighbors' preference to perform item-based CF. The user representation and the item representation are used to predict rating score. To evaluate the proposed method, GAN-HBNR on movieLen Small Latest datasets is compared with the proposed method to achieve higher accuracy.

Department: Mathematics and Computer Science Student's Signature ธันย์ชนก ตั้งปอง

Field of Study: Computer Science Advisor's Signature Saranya Maneeroj

Academic Year: 2019

ACKNOWLEDGEMENTS

First of all, I have acknowledged our senior project advisor, Associate Professor Saranya Maneeroj, Ph.D, for dedicating herself for this work. Without her assistance, this work would have never been achieved. Furthermore, she always supports me many things, not only working but also life attitude. She also inspires me with knowledge in the field of recommender systems.

I would like to appreciate my committee, including Associate Professor Vimolrat Ngamaramvaranggul, Ph.D, and Assistant Professor Kitiporn Plaimas, Ph.D, who have participated in my project proposal and senior project examinations. They suggested many important points in my project, and I hope that I will be able to improve my work as much as possible. Moreover, Assistant Professor. Dittaya Wanvarie, Ph.D, has supported AWS Cloud for GPU support in my work.

I also thank for financial support from the department of mathematics and computer science, faculty of science, Chulalongkorn University. I also would like to thank my friends for their support all over the time since I started my research. When I face the problems in implementation, suggestions from them are useful in debugging my work.

Moreover, I would like to thank my family for supporting me all over the time since I started my research including the suggestion how to survive on a difficult situation.

CONTENTS

CONTENTS (Cont.)

Page

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Background and Rationale

A recommender system (RS) is a system, which can predict and rank the future preference of a set of items for a user [1]. In general, recommendation lists are created based on user preferences, item features, user-item past Interactions [2]. RS is applied in several tasks, such as playlist recommenders for video and music services like Netflix, YouTube, and Spotify, product recommenders in Amazon, content recommenders in Facebook and Twitter [3]. Recommendation models are mainly categorized into content-based filtering (CBF) and collaborative filtering (CF) [2]

CBF is a method, which predicts items to a target user based on the feature of items, which was rated by the target user in the past. Since CBF only focuses on the target user's information, CBF only recommends items which are similar to the target user's items. So the items which are recommended by CBF are not varied and boring. This problem is called a serendipitous problem.

To solve this problem, CF is applied by using the information of neighbors who have a similar taste to the target user. So the items, which is recommended by CF, have more varieties. There are two main approaches in CF, which is user-based CF and item-based CF. User-based CF approach finds similar users of a target user and gives him/her recommendations based on what other people with similar consumption patterns appreciated [4]. While Item-based CF approach finds similarity between predicted items and sets of items, which are similar to the items that a target user gives a high rating in the past. The traditional user-based and item-based CF is to calculate similarity by using cosine similarity or Pearson correlation technique. Both user information and item information are important for RS. Thereby, my project focuses on user-based CF and item-based CF approaches.

There are several algorithms that are applied in CF and generative adversarial networks (GANs) [5] is also applied in CF. GANs is the state-of-the-art in image generation. It is applied in RS for generating rating scores and extracting better user representations and item representations. GANs is a deep generative model. It consists of two models which are trained under the adversarial learning idea including generator part and discriminator part. The generator part tries to generate fake data, which should be close to real data, while the discriminator part tries to learn representation of real data in order to classify real data and fake data. Therefore the discriminator part is used for learning user and item representations in CF.

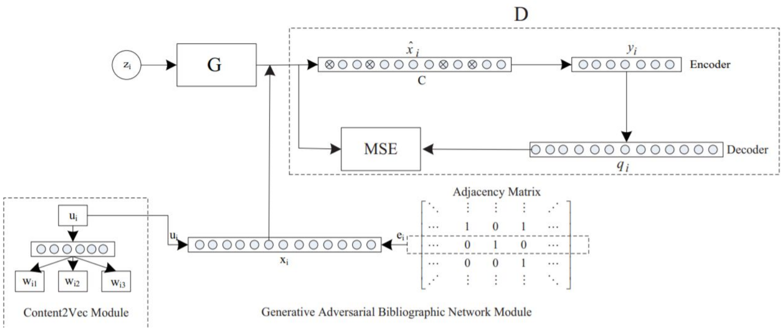

Figure 1.1 Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation [6]

The existing GAN recommendation is GAN Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation (GAN-HBNR) [6]. GAN-HBNR recommends the document related to the target user's document by calculating the similarity score from document representations and author representations. The document representations and author representations are extracted from the discriminator part in GANs. GAN-HBNR uses denoising autoencoder (DAE) as a discriminator part. DAE is an autoencoder, which randomly corrupts input to prevent overfitting problems. The input of GAN-HBNR (Figure 1) is the concatenation of two input vectors (x_i) which are the output of context2vec of the target document (u_i), which represents document embedding, and adjacency matrix of interaction between target document and authors (e_i).

From the input of GAN-HBNR, It only uses the information of the target user, while it ignores the opinions of other users or neighbors. It is common sense that the user's neighbors and item's neighbors are important to predict rating scores.

In my project, I focus on the CF method which is user-based and item-based CF by considering the opinions of user neighbors and item neighbors to predict the rating. By applying the CF method in my work, I add neighbor's preference into the input. In the case of user-based CF, I add preference of target user's neighbors who have the top highest similarity score with target user. While item-based CF, I add preference of target item's neighbors who have the top highest similarity score with target item. Furthermore, I use a transformer instead of DAE as the discriminator. DAE weighs every item, which the target user has rated with the same attention, despite the fact that items should not receive the same attention from the target user. I apply the transformer as a discriminator part. In the case of user-based CF, I use a self-attention mechanism to find similarities between target users and other users who rated corated items (The items who are rated by the same user) to simulate user-based CF for learning target user representation. On the other hand, I use transformers to find similarities between the target item and other items, which are rated by corated users (The users who rate the same items) to stimulate item-based CF for learning target item representation. Thereby, target user and target item representations are derived to represent the user's neighbors and item's neighbors.

To conclude, I propose a new recommendation system model to predict rating by using generative adversarial networks model on user preference, user neighbors' preference, and item neighbors's preference to perform user-based and item-based collaborative filtering. I compare the efficiency between the proposed model and the previous recommendation models by using accuracy.

1.2 Objectives

1. To propose a GAN-transformer model which incorporates transformers in a generative adversarial network-based model to learn user representation and item representation on user preference, user neighbors' preference and item neighbors's preference to perform user-based and item-based collaborative filtering in order to predict rating.
2. To use a transformer as discriminator in GAN in order to extract relation among users/items preferences and their neighbors's preference for learning users/items representation.
3. To compare the efficiency between the proposed model and the previous recommendation model by using accuracy.

1.3 Scope

1. I use the movieLen Small Latest Dataset. There are 100,836 ratings in the range of 0 to 5 (max rating is 5.0 and min rating is 0.5) from 610 users on 9,724 movies including userId, movieId, rating and timestamp.
2. I compare the efficiency of the proposed model with the research work of Xiaoyan Cai, Junwei Han, Libin Yang on Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation

1.4 Project Activities

1. Study research and academic articles related to recommendation systems.
2. Identify problems and disadvantages of previous work.
3. Design and analyze solutions to the problem of recommendation systems.
4. Implement and Improve the proposed model
5. Evaluate the proposed model.
6. Analyze and discuss the results in order to improve the proposed model in the future
7. Create related documents such as reports.

Table 1.1 Gantt chart of project activities

Procedure	2019					2020			
	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APL
1. Study research and academic articles related to recommendation systems.									
2. Identify problems and disadvantages of previous work.									
3. Design and analysis solutions to the problem of recommendation systems.									
4. Implement and Improve the proposed model									
5. Evaluate the proposed model.									
6.Analyze and discuss the results in order to improve the proposed model in the future									
7. Create related document such as reports.									

1.5 Benefits

1. Benefits in terms of knowledge and experience for the implementor.
    a. To get to learn the theory and operation of building new methods for recommendation systems.
    b. To get to apply knowledge in computer science to solve problems in the recommendation system.
2. Benefits in solving the social problem.
    a. To contribute the previous research in order to improve research in this field.
    b. To solve the problem of rating behavior of different users.
    c. To improve the recommendation system.

1.6 Report Outlines

The rest of this report is organized as follows:

1. Chapter 2 Literature Review: To present knowledge, related research
2. Chapter 3 Methodology: To explain the proposed method and its processes.
3. Chapter 4 Experimental evaluation: To discuss dataset, evaluation metrics including accuracy and experimental results.
4. Chapter 5 Conclusion: To discuss the efficiency of the proposed method and advantages compared with the previous method.

CHAPTER II

LITERATURE REVIEW

In this chapter, there are five main topic researches. First is collaborative filtering (CF) which is an approach in RS. It is categorized into user-based collaborative filtering (user-based CF), item-based collaborative filtering (item-based CF), Neural Collaborative Network including NCF, DeepICF and NCAE. Next, Generative Adversarial Networks (GANs) is introduced , Then, Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation (GAN-HBNR) which is the existing GAN recommendation is presented. After that, to extract representation in GAN-HBNR, Denoising autoencoder (DAE) is explained. However, the representation obtained from DAE is weighted with the same attention into all items, despite the fact that items should not receive the same attention from the target user. To weigh items with different attention from the target user, a transformer is introduced and applied in GAN instead of DAE.

2.1 Collaborative filtering (CF)

Collaborative Filtering is a popular approach in RS. It can filter out items that a user might like on the basis of reactions by similar users. There are three main techniques that deal with my work, which is user-based Collaborative filtering, Item-based Collaborative filtering and neural collaborative network.

2.2.1 User-based Collaborative filtering (User-based CF)

Traditional User-based CF is Memory-Based Collaborative Filtering, which uses user-item rating data to compute the similarity between users based on cosine similarity [7] and take a weighted average of ratings. The cosine similarity is computed by using Equation 1

$$similarity(A, B) = \frac{A \bullet B}{\|A\|\|B\|} \tag{1}$$

where A is a rating vector of target user A who rates to every item in the corpus.

B is a rating vector of user B who rates every item in the corpus.

User-based CF finds similar users to a target user and gives him/her recommendations based on what other people with similar consumption patterns appreciated. From the equation of user-based CF is

$$r_{ij} = \frac{\sum_{k} similarities(u_i,u_k)(r_{kj})}{\sum_{k} similarities(u_i,u_k)} \tag{2}$$

where r_{ij} is predicted rating for target user i toward target item j, u_i is target user, u_k is neighbor user, r_{kj} is rating, which neighbor has rated to item j. It indicates that user-based CF finds similarity between target user and other users who rated corated items.

Table 2.1 User-Item matrix[7]

Name	Avenger	Star wars	Thor	Spider-man	Iron Man
Alex	4	2	?	5	4
Bob	5	3	4	?	3
Tom	3	?	4	4	3

From table 2.1, I explain the step of user-based CF[7] for predicting the rating, which Alex rates to Thor. The step as follows:

Step 1 : Calculate the similarity score between Alex and all other users.

The calculation for the similarity between Alex and Bob can be derived from equation 1 as follows:

$$\text{similarity(Alex, Bob)} = \frac{(4\times5)+(2\times3)+(4\times3)}{\sqrt{4^2+2^2+4^2}\times\sqrt{5^2+3^2+3^2}} = 0.97\,.$$

The similarity score between Alex and Tom can be obtained by following the same as step 1.

Step 2 : Predict the ratings of Thor that are rated by Alex

Now, the movie Thor unrated by Alex can be calculated by using equation 2 as follows:

$$r_{Alex,Thor} = \frac{similarity(Alex,Bob)\times r_{Bob,Thor} + similarity(Alex,Tom)\times r_{Tom,,Thor}}{similarity(Alex,Bob)+similarity(Alex,Tom)} = 4.02\,.$$

2.2.2 Item-based Collaborative filtering

Traditional Item-based CF is Memory-Based Collaborative Filtering which uses item-user rating data to compute the similarity between item based on cosine similarity and a weighted average of ratings. Item-based CF finds similar items to a target item, which a target user gave a high rating in the past and then those items are given him/her recommendations based on what other items with similar scoring patterns. The equation of item-based CF is

$$r_{up} = \frac{\sum_{k} similarities(i,p)(r_{ui})}{\sum_{k} similarities(i,p)} \tag{3}$$

where r_{up} is predicted rating for target user u toward target item p, i is another item, which is similar to item p, and r_{ui} is rating that target user u rates for similar item i. It indicates that item-based CF finds similarity between target item and other items, which are rated by corated users.

From table 2.1, I explain the step of item-based CF [7] for predicting the rating that Alex rates to Thor. The step as follows:

Step 1 : Transpose the user-item matrix from table 2.1 to the item-user matrix as shown in table 2.2

Table 2.2 Item-User matrix [7]

	Alex	**Bob**	**Tom**
Avengers	4	5	3
Star wars	2	3	?
Thor	?	4	4
Spider-man	5	?	4
Iron Man	4	3	3

Step 2 : Calculate the similarity score between Thor and all other items.

The calculation for the similarity between Thor and Avengers can be derived from equation 1 as follows:

$$\text{similarity(Thor, Avengers)} = \frac{(4\times5)+(4\times3)}{\sqrt{4^2+4^2}\times\sqrt{5^2+3^2}} = 0.97$$

The similarity score between Thor and all other items can be obtained by following the same way, which is 0,0,1 respectively.

Step 3 : Predict the ratings of Thor that are rated by Alex

Now, the movie Thor unrated by Alex can be calculated by using equation 3 as follows:

$$r_{Alex,Thor} = \frac{similarity(Thor,Avenger)\times r_{Alex,Avengers} + similarity(Thor,\ Iron\ man)\times r_{Alex,,Iron\ man}}{similarity(Thor,Avengers)+similarity(Thor,\ Iron\ man)} = 4.0$$

2.2.3 Neural Collaborative Network

Neural Collaborative Network is CF which uses a neural network to learn the complex structure of user-item interaction. Such as neural collaborative filtering (NCF), Deep item-based collaborative filtering (DeepICF) and Neural Collaborative Autoencoder (NCAE).

2.2.3.1 NCF

NCF [8] is a neural network to predict the rating score that target user *u* gives to target item *i*. From figure 2.1, the input of NCF is a target user *u* one-hot vector and a target item *i* one-hot vector and then feed them into the embedding layer, which is q_1 (user embedding vector) and p_2 (item embedding vector). After that feed q_1 and p_2 into the interaction hidden layer for learning the relation between user and item. Finally, the latent feature vector from the interaction hidden layer predicts rating score $\widehat{y_{ui}}$. The equation is below:

$$\widehat{y_{ui}} = f(P^T v_u^U\ ,\ Q^T v_i^I \mid P\ , Q\ , \Theta_f\ ) \tag{4}$$

where P and Q, denoting the latent factor matrix for users and items, respectively; and Θ_f denotes the model parameters of the interaction function f. Since the function f is defined as a multi-layer neural network, it can be formulated as

$$f(P^T v_u^U\ ,\ Q^T v_i^I\ ) = \phi_{out}(\phi X(\ldots\phi_2(\phi_1(P^T v_u^U\ ,\ Q^T v_i^I\ ))..)) \tag{5}$$

where ϕ_{out} and ϕX respectively denote the mapping function for the output layer and x-th neural collaborative filtering (CF) layer, and there are X neural CF layers in total

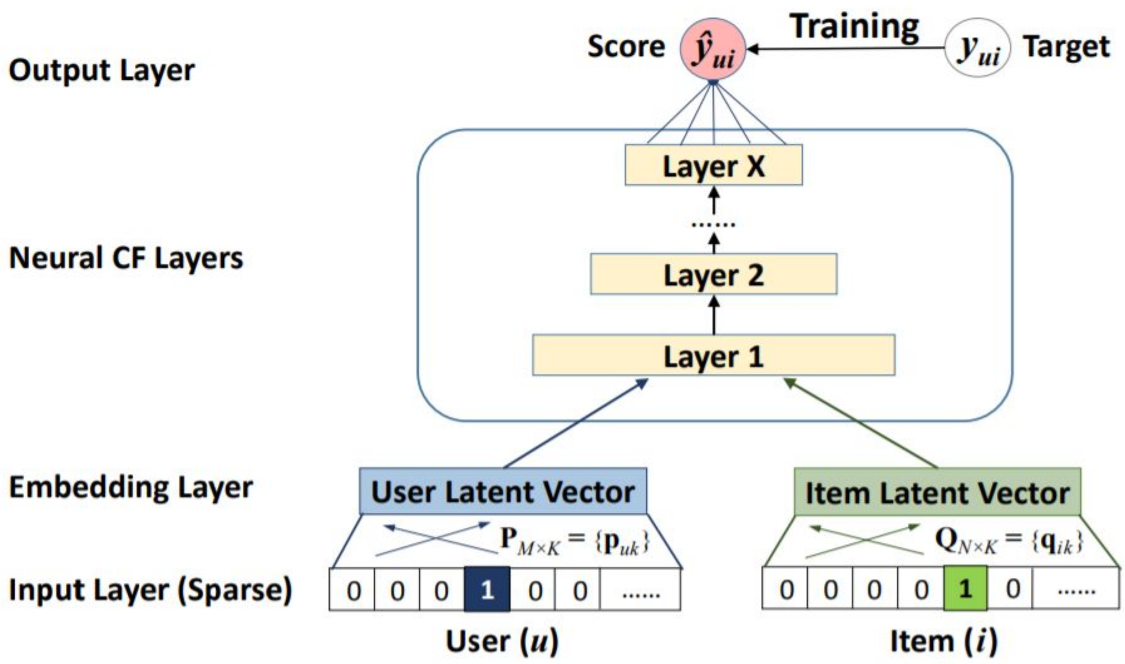

Figure 2.1 Architecture of NCF [8]

2.2.3.2 DeepICF

DeepICF [9] is also the neural network, which predicts the rating score that target user u give to target item i. But the structure of DeepICF is more complex. From figure 2.2, the input of DeepICF a muli-hot vector which represents any users who rated target item i and a target item i one-hot vector and then feed them into the embedding layer which is q_1 ,q_3 ,q_n (user embedding vectors) and p_2 (item embedding vector). And then element-wise each pair of user and item embedding vector in pairwise interaction layer. After that each pairwise vector is pooled into one vector in pooling layer. The weighted average pooling used in DeepICF is defined as follows:

$$f_{avg}(V_{ui}) = \frac{1}{(|\Re_u^+|-1)^{\alpha}}\left(\sum_{j\in\Re_u^+\setminus i} q_j \odot p_i\right) \tag{6}$$

where α is the normalization hyper-parameter that controls the smoothing on V_{ui} of different sizes.

The following pooled vector is fed into the deep interaction layer for learning the relation between user and item. They give the formal definition of the deep interaction layers as follows:

$$e_1 = ReLU(W_1 e_{ui} + b_1) \tag{7}$$
$$e_2 = ReLU(W_2 e_1 + b_2)$$
$$\cdots$$
$$e_L = ReLU(W_L e_{L-1} + b_L)$$

where W_l , b_l , and e_l denote the weight matrix, bias vector, activation function and output vector of the $l-th$ hidden layer, respectively

Finally, the latent feature vector from interaction hidden layer predict rating score $\widehat{y_{ui}}$ The equation as follows:

$$\widehat{y_{ui}} = z^{\top} e_L + b_u + b_i \tag{8}$$

where z , b_u and b_i denotes the weight vector, user bias, and item bias of the prediction layer, respectively.

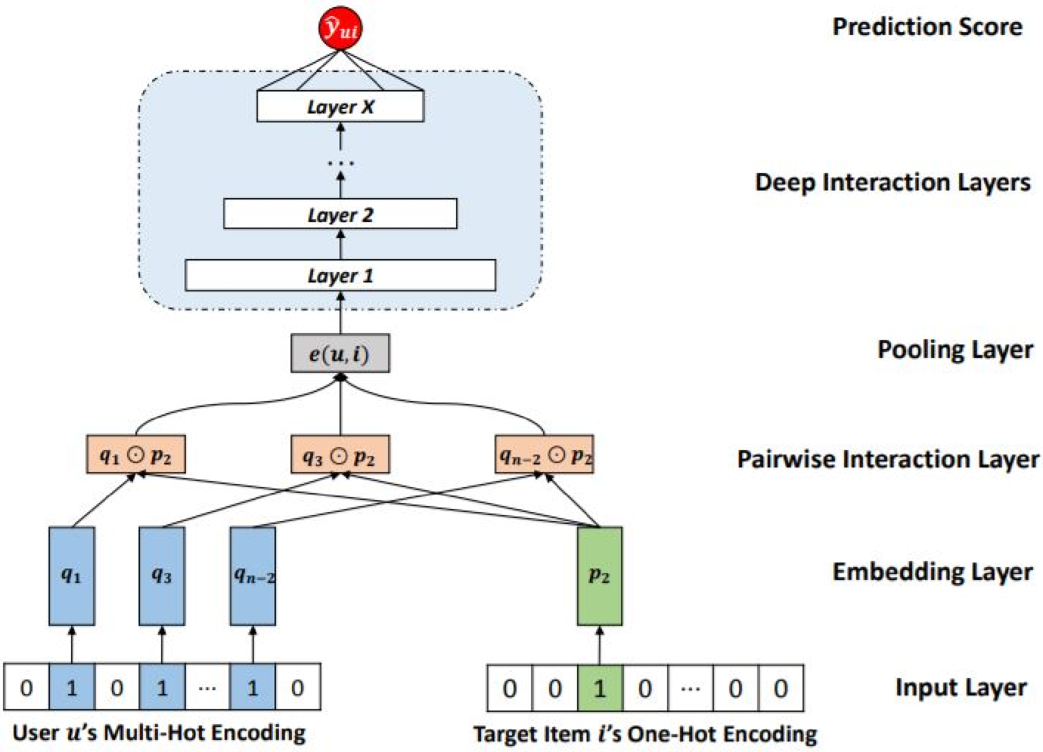

Figure 2.2 Architecture of DeepICF [9]

2.2.3.3 NCAE

NCAE [10] is an autoencoder (the generative neural network) that compresses user-item rating input into low dimensional vector and NCAE tries to reconstruct that low dimensional vector into the original input. So NCAE is used to learn the representation of users and items. From figure 2.3, The input of NCAE is a row of user-item rating matrix which represents the rating score that the user *i* gives to whole items in the corpus. The input of NCAE is corrupted with noise in order to prevent overfitting before training. The corrupted input is fed into the encoding layer to a low dimensional vector and then reconstructed that low dimensional vector into the original input in the decoding layer. After training, NCAE uses that low dimensional vector as user representation. NCAE is defined as:

$$\widehat{u_i} = nn(\tilde{u}_i) = z_i^L = \phi^L(W^L(\ldots\phi^1(\phi_1(W^1\tilde{u}_i + b^1)..) + b^L) \tag{9}$$

where the first *L-1* layer of NCAE aims at building a low-rank representation for user *i*, and the last layer *L* can be regarded as item representation learning or a task-related decoding part.

NCF, DeepICF, and NCAE focus only on the user-item interaction while they ignore user-user interactions or item-item interactions from the corated data which is the core of the CF-based approach.

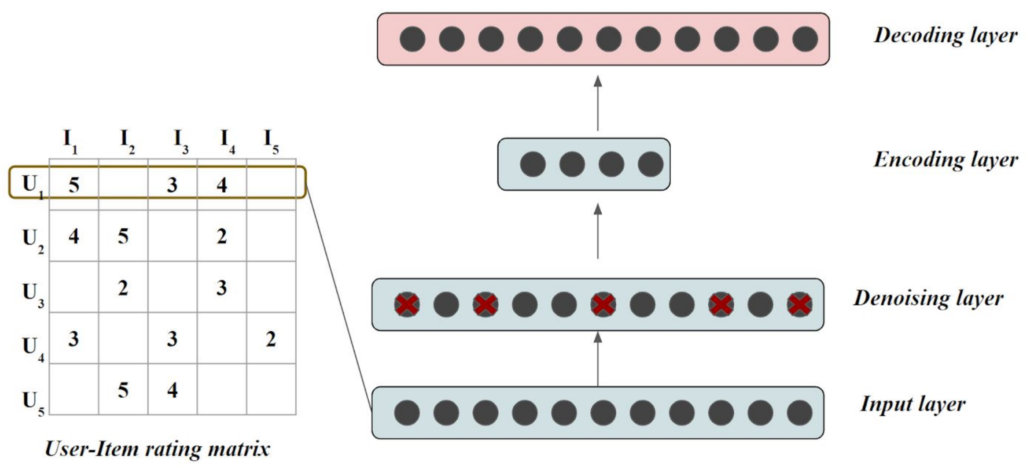

Figure 2.3 Architecture of NCAE

2.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) [11] is the generative model that combines two neural networks, that work in adversarial ideas. From figure 2.4, first is discriminator, it tries to discriminate between real data and fake data. Second is the generator, it tries to generate fake data to be closed to real data. In discriminator function, it should be 1 if it is real data in contrast it should be 0 if it is fake data. GANs train the discriminator and generator until the generator generates fake data, which can fool the discriminator. So the objective function of GANs is to maximize discriminator loss(D) and minimize generator loss(G). D and G play the following two-player minimax game with value function $V(G, D)$:

$$min_G max_D V(D, G) = \Xi_{x\text{-} p_{data(x)}}[log\ D(x)] + \Xi_{x\text{-} p_{z(z)}}[1 - D(G(z)))] \quad (10)$$

Where:

$D(x)$ is the discriminator's estimate of the probability that real data instance x is real.

E_x is the expected value over all real data instances

$G(z)$ is the generator's output from input noise z.

$D(G(z))$ is the discriminator's estimate of the probability that a fake instance is real.

E_z is the expected value over all random inputs to the generator

Since GAN is good at capturing the distribution of data by using the implicit function to learn the complex distribution of data in polynomial time, GANs can generate data that is close to real. So GANs are adopted to learn representation tasks. Therefore I proposed the GANs model to extract better user and item representations in RS.

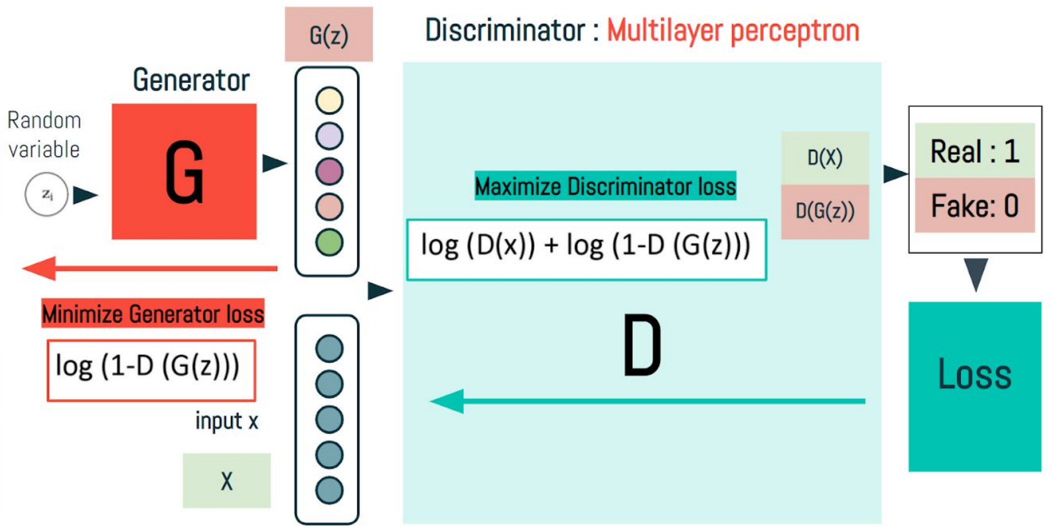

Figure 2.4 Architecture of GANs

2.3 Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation (GAN-HBNR)

GAN-HBNR [6] is a Personalized Citation Recommendation system that applies GANs to learn document and author representation in the discriminator part. GAN-HBNR uses DAE as a discriminator part. Document and author representation are derived to calculate similarity scores, a document with a high similarity score is recommended to the user. From figure 2.5, the input of GAN-HBNR (x_i) is concatenation between a vector u_i which is an embedded vector by using the content2vec module from i^{th} document/author text information and a vector e_i which is i^{th} row of the document-author interaction matrix. Input x_i is fed into the discriminator to learn the representation of the document and author and then the generator generates *G(Z)* from random noise *z* in order to feed *G(Z)* to the discriminator. Normally, Discriminator answers 1 if the input is x_i but answers 0 if the input is *G(Z)*. The document and author representations are extracted when the generator generates *G(Z),* which can fool the discriminator. After training in several steps, They get a hidden feature representation vector of the document and author. These vectors is used to calculate the similarity score as follows:

$$r_q = V_{PR} v_{q_t}^T + V_{AR} v_{q_a}^T \tag{11}$$

where r_q is similarity score, V_{PR} is the vector representation of training documents, v_{q_t} is the vector representation of the testing document, V_{AR} is the vector representation of authors related to training documents, v_{q_a} is the vector representation of the authors related to the testing document.

From the input of GAN-HBNR, it is CBF, because it only focuses typically on information from the target user or target item for creating user/item representation while ignoring the opinions of other users or neighbors. It is common sense that the user's neighbors and item's neighbors make a significant effect on the user's characteristics and item's characteristics. Therefore I proposed a new GANs method based on information of the user's neighbor and item's neighbor.

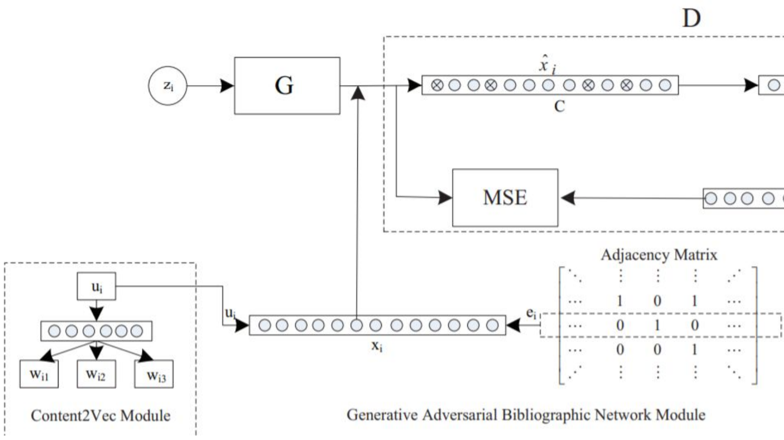

Figure 2.5 Architecture of GAN-HBNR

2.4 Denoising autoencoder(DAE)

DAE [12] is an autoencoder that is trained by corrupted input in order to solve overfitting problems. From figure 2.6, input X is corrupted by noise. The corrupted input $\tilde{x}$ is mapped to a hidden representation with the same process of the standard autoencoder as follows:

$$h = f_\theta(\tilde{x}) = s(W\tilde{x} + b) \tag{12}$$

Where h is a hidden representation, $\tilde{x}$ is corrupted input, W and b are weight and bias respectively. From the hidden representation the model reconstructs as follows:

$$z = g_{\theta'}(h) \tag{13}$$

where z is reconstructed input. When reconstructed input is closed to original input, DAE can use hidden representation instead of original input. So DAE is used in RS by extracting user and item representation.

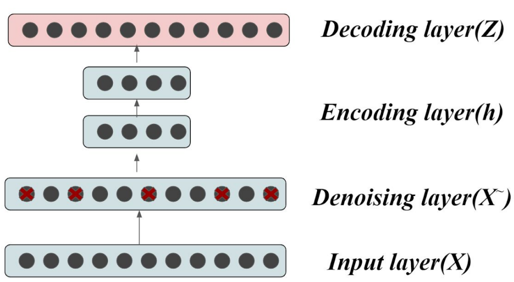

Figure 2.6 Architecture of DAE

2.5 Transformer

Transformer [13] is used primarily in the field of natural language processing (NLP). It is used for machine translation. The transformer consists of two main components, which are a set of encoders chained together and a set of decoders chained together. There are four main parts in the transformer, which are Encoder and Decoder Stacks, Attention, Position-wise Feed-Forward Networks and Positional Encoding as shown in figure 2.7.

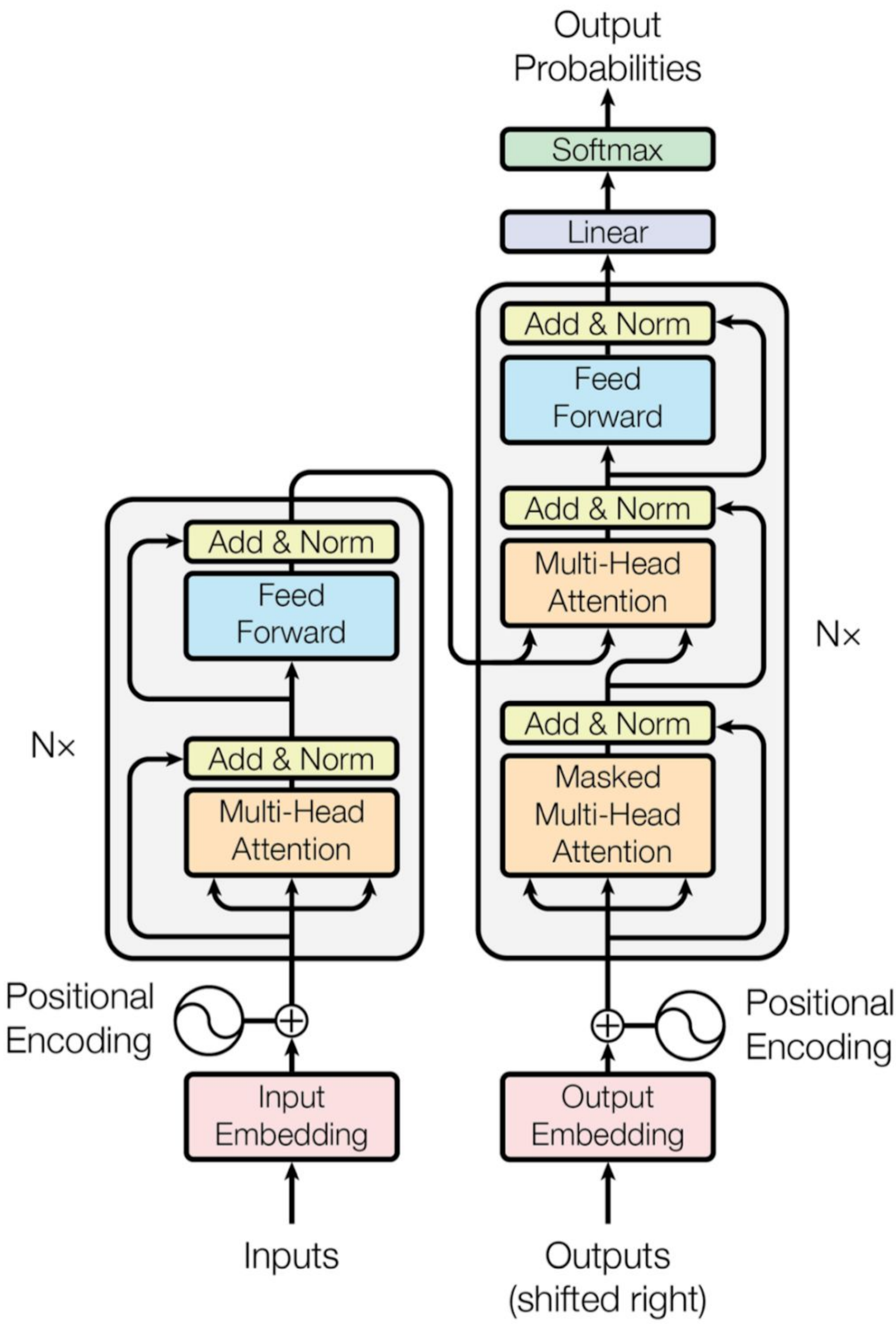

Figure 2.7 Architecture of Transformer[13]

2.5.1 Encoder and Decoder Stacks

Encoder: The encoder is composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple position-wise fully connected feed-forward network.

Decoder: Similar to the encoder, the decoder is also composed of a stack of $N = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

2.5.2 Attention

The highlight of a transformer is a self-attention mechanism, which learns the attention between target word and other words in sequence. The attention function is called Scaled Dot-Product Attention. It computes by using Equation 14.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{14}$$

where, Q is queries (vector representation of one word in the sequence input) , K is keys (vector representations of all the words in the sequence input), d_k is dimension of keys and V is values (vector representations of all the words in the sequence)

Instead of performing a single attention function, they apply several attention functions to learn different representations by concatenating all representations into one as follows:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{15}$$

$$\text{Where } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

2.5.3 Position-wise Feed-Forward Networks

In addition to the sub-layers of attention, each layer in encoders and decoders has a fully connected forward feed network for separating and identifying each position.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{16}$$

2.5.4 Positional Encoding

Since the transformer contains no recurrence and no convolution, in order to use the order of the sequence, it must inject some information about the relation of the tokens in the sequence. There are many ways of positional encodings. In this work, they use sine and cosine functions of different frequencies as follows:

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{17}$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{18}$$

where pos is the position and i is the dimension. That is each dimension of the positional encoding correlates with a sinusoid.

In my model, I incorporate the transformer to find attention between target users and other users who rated corated items to simulate user-based CF and to find attention between the target item and other items, which are rated by corated users to simulate item-based CF.

CHAPTER III
METHODOLOGY

In this chapter, I proposed a new method that applied GAN with the transformer on both user-based collaborative filtering in order to extract user representation, which represents the user's neighbors and item-based collaborative filtering in order to extract item representation, which represents the item's neighbors. After that, I predict the rating scores from the derived user representation and item representation by adopting a neural network-based approach. There are three main steps in my model, which is input preparation, representation learning, and rating prediction as shown in figure 3.1.

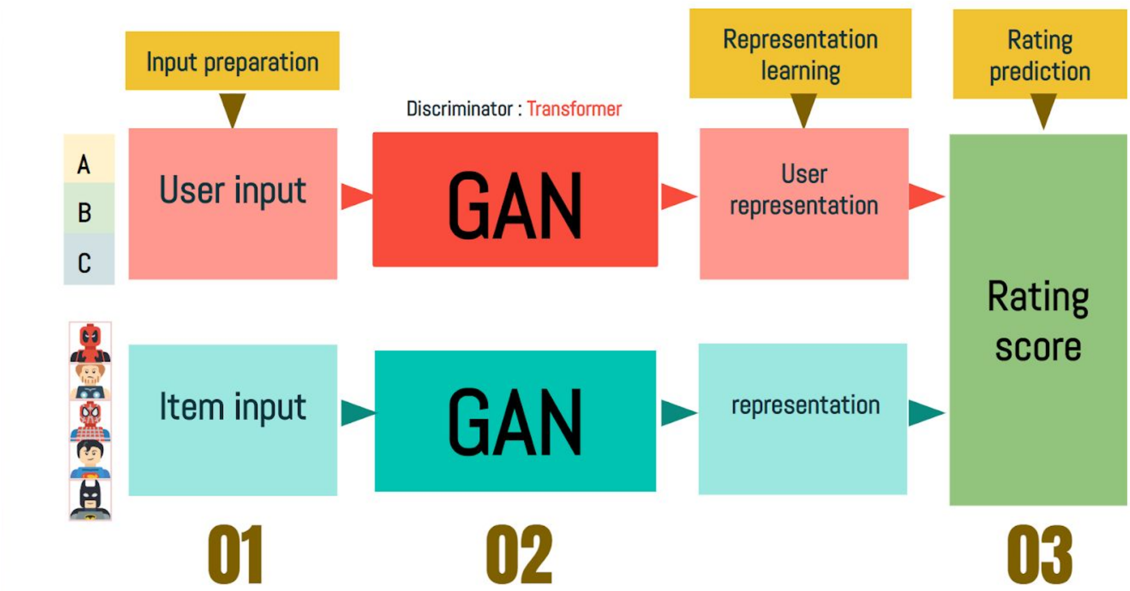

Figure 3.1 the overview of proposed model

3.1 Input preparation

In this section, I prepare input of user sequences and item sequences. For the user sequence input, it represents information of the target user and his/her neighbors. For the item sequence input, it represents information of the target item and its neighbors.

3.1.1 User sequence input

I prepare target user sequence input called UI_i which is a sequence of a row of target user *i* and rows of target user *i's* neighbors from the user-item rating matrix which store rating that target user *i* has rated every items in the corpus. For target user *i*'s neighbors, I pick neighbors from any users who have rated with the same item and also have the highest top *n* similarity score to target user *i*. The similarity score is considered from the user similarity matrix.

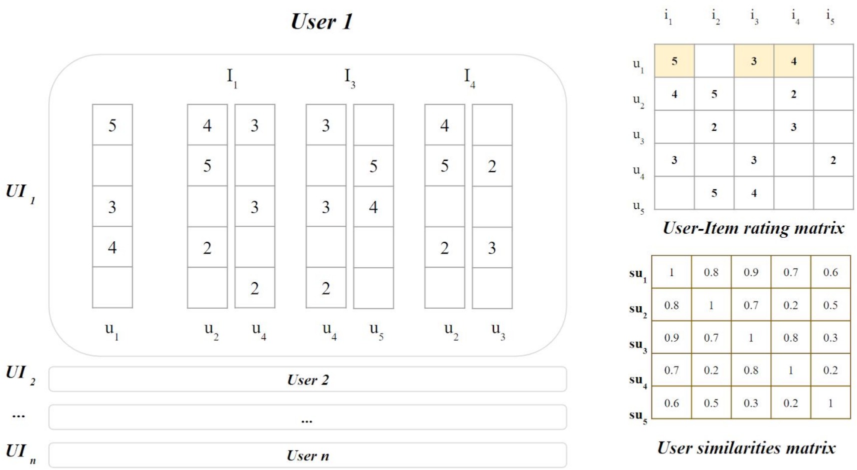

Figure 3.2 User sequence input preparation

For example, I want to prepare user1 sequence input called UI_1 . The step as follows:

1. In the user-item matrix, I consider the row of user1 (u_1) which has items that user1 has rated. That is item1 item3 and item4
2. For item1, I pick other users who also have rated item1 in the corpus. And then I pick user 1'neighbors who have the highest top *n* similarity score to user1 (the similarity is considered from user similarity matrix), which is user2 and user 4
3. Repeat step 2 for item3 and item4. (In case of item3, user 1'neighbors are user4 (u_4) and user5 (u_5). In case of item4, user 1'neighbors are user2 (u_2) and user3 (u_3))
4. For all rows of user 1's neighbors ($u_2, u_4, u_4, u_5, u_2, u_3$) who have rated item1 item2 and item3 and row of user1 (u_1) from the user-item matrix, I combine them together in order to form user1 sequence input, which represents information of user1 and his/her neighbors.

3.1.2 Item sequence input

Similar to the user input preparation, I also prepare in terms of sequence but instead of preparing from the user-item matrix, I prepare from the item-user matrix, which stores a rating that the target item *j* is rated by every users in the corpus.

The target item sequence input called IU_j, which is a sequence of a row of target item *j* and rows of target item *j*'s neighbors from the item-user rating matrix.

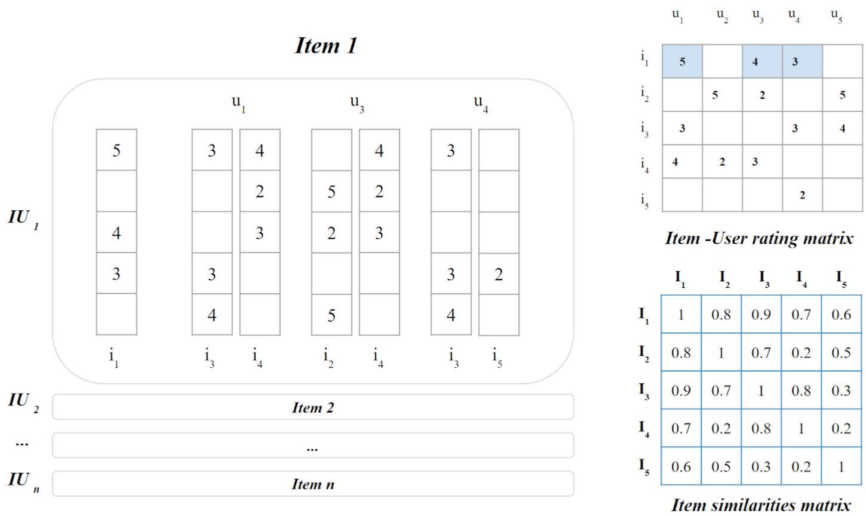

Figure 3.3 Item sequence input preparation

For example, I want to prepare item1 sequence input called IU_1. The step as follows:

1. In item-user matrix, I consider the row of item1 (i_1), which has users that item 1 is rated. That is user1 user3 and user4
2. For user1, I pick other items which also is rated by user1 in the corpus. And then I pick item1's neighbors which have the highest top *n* similarity score to item1 (the similarity is considered from user similarity matrix), which is item3 and item4
3. Repeat step 2 for the user3 and user4 (In case of user3, item1's neighbors are item2 (i_2) and item4 (i_4). In case of user4, item1's neighbors are item3 (i_3) and item5 (i_5))
4. For all rows of item 1's neighbors ($i_3, i_4, i_2, i_4, i_3, i_5$) which is rated by user1, user3,user4 and a row of item1 (i_1) from item-user matrix, I combine them together in order to form item1 sequence input, which represents information of item1 and its neighbors.

Finally, I have prepared the target user sequence input (UI_i) and the target item sequence input (IU_j) to use for learning target user representation and target item representation in the next section.

3.2 Representation learning

In this section, I propose a new method which applies GAN with a transformer on both user-based CF and item-based CF in order to extract user representation and item representation respectively. The proposed model is called GAN-transformer. I feed the input of user sequences (UI_i) into a user GAN-transformer, which is for learning user representation, which represents the user's neighbors. On the other hand, I feed the input of item sequences(IU_j) into item GAN-transformer which is for learning item representation, which represents item's neighbors. The structure of the user GAN-transformer and item GAN-transformer are quite similar. But only different in terms of input preparation. Therefore, I only explain the work of the GAN-transformer as shown in figure 3.4.

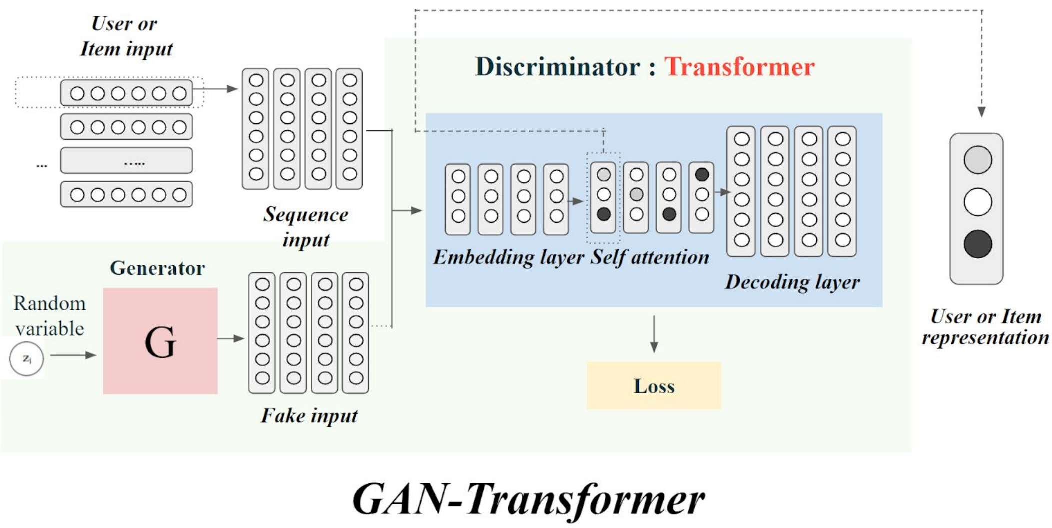

Figure 3.4 Architecture of GAN-Transformer

3.2.1 Embedding layer

I feed prepared input through the embedding layer to embed them into low dimension as equation

$$e_i = f(W_e\, input + b_e) \tag{19}$$

where e_i, W_e and b_e are embedding vectors, the weight of embedding layer and bias respectively.

3.2.2 Self-attention layer

I feed e_i into the Self-attention layer in order to find the relation between target user/item and target user/item's neighbors by using attention technique in the transformer as follows:

$$A_i = softmax(\frac{Q_i K^T}{\sqrt{d_k}})V_i + e_i \tag{20}$$

where A_i is the output of self-attention layer , Q_i is queries(vector representation of target user/item in the sequence input) and K is keys (vector representations of all the users/items in the sequence input) d_k is embedding size and V_i is the values (vector representations of all the users/items in the sequence input)

3.2.3 Decoding layer

I feed A_i into decoding layer in order to reconstruct them to output, which should be close to the input as the equation

$$y_i = g(W_d A_i + b_d) \tag{21}$$

where y_i , W_d and b_d are the output of the decoding layer, the weight of the decoding layer and bias respectively.

3.2.4 Discriminator loss

I calculate loss between input and output of the decoding layer by the mean squared reconstruction error as the discriminator loss :

$$D\,(input) \;=\; \frac{1}{length\ of\ sequence\ input}(input_i - y_i)^{\,2} \tag{22}$$

3.2.5 GAN loss

For real input and fake input (generated by generator), I optimize generator and discriminator by pushing down on the discriminator value of real input, and pushing up on the discriminator value of fake input.

$$L_D\,(real\ input_i, Z_i) = D\,(real\ input_i) + [1 - D\,(G\,(Z_i))]^{\,+} \tag{23}$$

$$L_G\,(Z_i) = D\,(G\,(Z_i)) \tag{24}$$

Where L_D is discriminator loss

L_G is generator loss

$G\,(Z_i)$ is generated input, which is generated from random noise Z_i

After several training sessions, I pick user representation and item representation from the self-attention layer at the position of target user/item in the sequence, which has already represented the user's neighbor and item's neighbor in order to predict rating scores in the next section.

3.3 Rating prediction

As described above in section 3.1 and 3.2, I receive target user representations as A_{u_i} and target item representations as A_{I_j} respectively. In the prediction state, the rating of user u_i towards item I_j is computed by

$$\widehat{r}_{u_i,I_j} \;=\; A_{u_i}^{T}A_{I_j} + b_{u_i} + b_{I_j} \;+ \alpha \tag{25}$$

where b_{u_i}, b_{I_j} and α are bias of target user u_i, bias of target item I_j and global bias. I also use $L2$-norm for loss function, which can be defined as

$$L = \sum(r_{u,I} - \widehat{r}_{u,I})^{2} \tag{26}$$

where $r_{u,I}$ is the true rating and $\hat{r}_{u,I}$ is the rating prediction

CHAPTER IV

EXPERIMENTAL EVALUATION

In this section , experimental results of the proposed method, which focuses on the user's neighbors and item's neighbors information are presented by comparing with the GAN-HBNR, which only focuses on user and item information. This section is organized as follows. First, details of the datasets, which are used in these experiments. Second, evaluation metrics in this work are evaluated by NDCG . Finally, experiment results of GAN-HBNR and the proposed method is compared.

4.1 Dataset

A movieLen Small Latest Dataset[14] was used to evaluate the proposed method. It contains 100,836 ratings from 610 users on 9,724 items with rating range from 0.5 to 5.0. From 610 users on 9,724 items, I only picked users who have rated over 100 items and items, which are rated by more than 20 users. Therefore, I extract 157 users on 2,121 items with rating range from 0.5 to 5.0. The details of this dataset are shown in Table 4.1. This dataset is separated into a training set 80%, and test set 20% of the following ratings. This dataset provides userId, movieId, ratings and timestamp. The sample of this dataset used in these experiments is shown in Table 4.1.

Table 4.1 The sample from movieLen Small Latest Dataset [14]

userId	movieId	rating	timestamp
1	1210	5.0	964980499
1	2018	5.0	964980523
1	2628	4.0	964980523
1	2826	4.0	964980523
1	3578	5.0	964980668

From the described dataset, it can be summarized about basic information as shown in Table 4.2.

Table 4.2 Basic Information Summary of the Dataset

Characteristic	Dataset
Number of users	157
Number of items	2,121
Number of rating records	54,469
Rating range	0.5-5.0

From the dataset, I prepare data for learning user and item representation in my proposed model in the form of sequence input. The sequence inputs are divided into two parts, which are user sequence input and item sequence input.

For user sequence input, I sorted the first 100 items which target user has rated according to the timestamp in descending order. For each item, I pick the user's neighbors who have the top 3 similarity score to the target user and also have rated that item. Finally, I get the target user input, which consists of 300 target user's neighbors and target user information.

Similar to user sequence input, I prepare item sequence input by sorting the first 20 users who have rated the target item according to the timestamp in descending order. For each user, I pick the item's neighbors, which have the top 3 similarity score to the target item and also are rated by that user. Finally, I get the target item input, which consists of 60 target item's neighbors and target item information.

4.2 Evaluation metrics

There are many ways to evaluate the performance of the recommender system.
In this work, I evaluate the performance of the models by using Normalized Discounted Cumulative Gain (NDCG) [15] that is used to evaluate the quality of a recommended rank list on user preference. NDCG calculates how good of the recommendation lists which each

item on recommendation lists has a relevance score. Let K is the number of items on the recommendation list. $NDCG$ can be computed by using Equation (27).

$$NDCG_k = \sum \frac{DCG_{uk}}{IDCG_{uk}} \tag{27}$$

where DCG is discounted cumulative gain, which can be calculated as Equation (28) and $IDCG$ is ideal discounted cumulative gain, which is the highest DCG value among the possible ranked item list.

$$DCG_{uk} = \sum_{j=1}^{k} \frac{2^{r_{uj}}-1}{log_2(1+j)} \tag{28}$$

For explaining this evaluation, Table 4.2 is sorted by actual rating and predicted rating as shown in Table 4.3. Let $K=4$, DCG, and $IDCG$ of this user can Item Actual rating Predicted rating be computed as follows. Rating 2.0, 3.5, 5.0 and 4.0 in DCG denotes predicted rating of the item which has the top-4 highest rating in actual rating including item 1, 3, 5 and 7. Moreover, rating 5.0, 4.0, 3.5 and 3.0 denotes predicted rating which has the top-4 highest rating in predicted rating including item 5, 7, 3 and 8.

Table 4.3 Actual and predicted rating of the user u_1 [16]

Item	Actual rating	Predicted rating
1	5.0	2.3
2	-	2.4
3	4.0	3.5
4	-	1.0
5	4.0	5.0
6	-	2.0
7	3.5	4.0
8	1.5	3.0

Table 4.4 Sorted actual and predicted rating of the user u_1 [16]

Sorted item	Actual rating	Sorted item	Predicted rating
1	5.0	5	5.0
3	4.0	7	4.0
5	4.0	3	3.5
7	3.5	8	3.0
8	1.5	2	2.4
2	-	1	2.3
4	-	6	2.0
6	-	4	1.0

$$DCG_{u4} = \frac{2^{4.0}-1}{log_2(1+1)} + \frac{2^{3.5}-1}{log_2(1+2)} + \frac{2^{4.0}-1}{log_2(1+3)} + \frac{2^{1.5}-1}{log_2(1+4)} = 27.79$$

$$IDCG_{u4} = \frac{2^{5.0}-1}{log_2(1+1)} + \frac{2^{4.0}-1}{log_2(1+2)} + \frac{2^{4.0}-1}{log_2(1+3)} + \frac{2^{3.5}-1}{log_2(1+4)} = 52.41$$

$$NDCG_{u4} = \frac{DCG_{u4}}{IDCG_{u4}} = \frac{27.79}{52.41} = 0.568$$

4.3 Experiment results

To evaluate my proposed method, I compare it with GAN-HBNR, which does not consider preference on the user's neighbors and the item’s neighbors. In these experiments, the assumption that the proposed method, which extends from GAN-HBNR by considering neighbors preferences of the user will provide more accuracy (NDCG), will be proved. I separated the dataset into a training set 80%, and a test set 20%. I also use the same test set to evaluate the proposed model and GAN-HBNR. The experimental results are shown in Table 4.5 and figure 4.1

Table 4.5 Comparison of NDCG @K between proposed model and GAN-HBNR

NDCG @K	Proposed model	GAN-HBNR
K=5	0.6431	0.5246
K=10	0.6556	0.5418
K=15	0.6780	0.5704
K=20	0.7008	0.5986
K=22	0.7097	0.6096

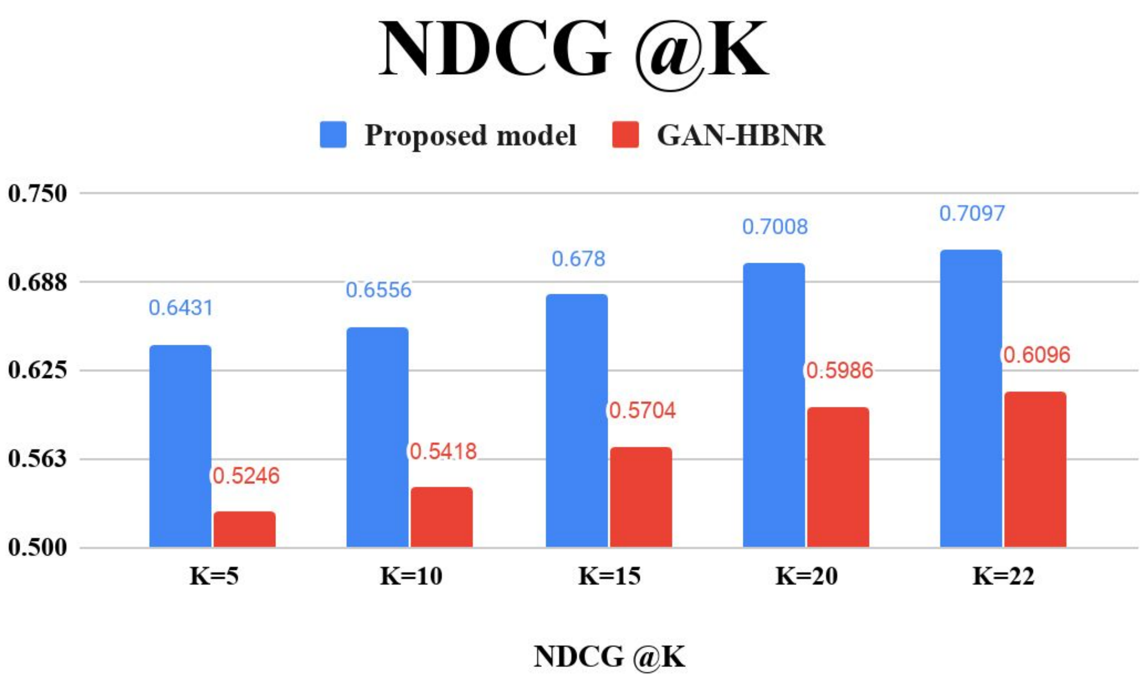

Figure 4.1 Comparison of NDCG@K between proposed model and GAN-HBNR

Figure 4.1 and Table 4.4 show my proposed model provides more accuracy (NDCG) than GAN-HBNR. Because GAN-HBNR uses only information of target user which is ratings that target user has rated all items in dataset for extracting target user representation and information of target item, which is ratings that target item is rated by all users in dataset for extracting target item representation. while my proposed model used both information of target user and target user’s neighbors for extracting target user representation and information of target item and target item’s neighbors for extracting target item representation. I can extract relations between users/items and their neighbors by using a transformer, which gives us better users/items representation than GAN-HBNR’s users/items representation.

CHAPTER V
CONCLUSION

This work proposed a GAN-transformer, which incorporates a transformer with GAN for user-based and item-based CF. In terms of user-based CF, I can extract relations of preference among target user and his/her neighbors by using a transformer as a discriminator in GAN. In terms of item-based CF, I can extract relations of preference among target item and its neighbors by using a transformer as a discriminator in GAN as well. From the experimental results in the previous chapter, it shows considering information of neighbors gets better accuracy than ignoring them. Therefore my proposed model gives better results than GAN-HBNR.

References

[1] Rishabh Mall.Recommender System[internet].2019[cited 2018 Oct 20]Available from:https://towardsdatascience.com/recommender-system-a1e4595fc0f0

[2] Zhang, Shuai et al. “Deep Learning Based Recommender System.” ACM Computing Surveys 52.1 (2019): 1–38. Crossref. Web.

[3] Wikipedia contributors. (2019, October 21). Recommender system. In Wikipedia, The Free Encyclopedia. Retrieved 06:54, October 22, 2019, from https://en.wikipedia.org/wiki/Recommender_system

[4] PETER, B., MELKER, F. Comparison of User Based andItem Based Collaborative Filtering.STOCKHOLM, SVERIGE 2017, 2017.

[5] Wang, Kunfeng & Gou, Chao & Duan, Yanjie & Yilun, Lin & Zheng, Xinhu & Wang, Fei-Yue. (2017). Generative Adversarial Networks: Introduction and Outlook. ACTA AUTOMATICA SINICA. 4. 588-598. 10.1109/JAS.2017.

[6] Cai, Xiaoyan, Han, Junwei, AND Yang, Libin. "Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation" AAAI Conference on Artificial Intelligence (2018): n. pag. Web. 22 Oct. 2019

[7] 王斌.Comparison of User-Based and Item-Based Collaborative Filtering[internet]. 2018[cite 2020 March 3] Available from:https://medium.com/@wwwbbb8510/comparison-of-user-based-and-item-based-collaborative-filtering-f58a1c8a3f1d

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW ’17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. DOI:https://doi.org/10.1145/3038912.3052569

[9] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep Item-based Collaborative Filtering for Top-N Recommendation. ACM Trans. Inf. Syst. 37, 3, Article 33 (April 2019), 25 pages. DOI:https://doi.org/10.1145/3314578

[10] Li, Qibing and Xiaolin Zheng. “Deep Collaborative Autoencoder for Recommender Systems: A Unified Framework for Explicit and Implicit Feedback.” ArXiv abs/1712.09043 (2017): n. Pag.

References (Cont.)

[11] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David,Ozair,Sherjil,Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.

[12] Vincent, Pascal; Larochelle, Hugo (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". Journal of Machine Learning Research. 11: 3371–3408.

[13] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. “Attention is All you Need.” NIPS (2017).

[14] Grouplens.MovieLens[internet].2019[cite 2019 December 15]Available from:https://grouplens.org/datasets/movielens/

[15] Stanford(2013).Introduction to Information Retrieval[PDF file]. Retrieved from https://web.stanford.edu.

APPENDIX

APPENDIX A

The Project Proposal of Course 2301399 Project Proposal

Academic year 2019

Project name	Incorporate Transformers with Generative Adversarial Networks for User-based and Item-based Collaborative filtering recommendation
Advisor	Assoc. Prof. Dr. Saranya Maneeroj. Ph.D.
Provider	Thunchanok Tangpong Student identification number 5933632323
	Department of Mathematics and Computer Science
	Faculty of Science, Chulalongkorn University

Rational Criterion

A recommender system is a system that can predict and rank the future preference of a set of items for a user [1]. In general, recommendation lists are created based on user preferences, item features, user-item past Interactions [2]. Recommender systems are applied in several tasks, such as playlist recommenders for video and music services like Netflix, YouTube and Spotify, product recommenders in Amazon, content recommenders in Facebook and Twitter [3]. Recommendation models are mainly categorized into collaborative filtering, content-based recommender system and hybrid recommender system [2].

Content-based filtering is a method based on user profile and item profile to recommend the items to the users. It recommends items that are similar to those that a user liked in the history. Content-based recommender tries to solve user preference problems based on user-item interactions.

Collaborative Filtering (CF) is a method that can filter items that users may like by using similar user's reactions [4]. CF is a well-known and successful method in personalized recommendation systems. CF recommendation purpose is to calculate a list of interesting items to target users based on the preferences of their neighbors (neighborhood users with similar opinion) [5]. There are user-based CF and item-based CF. Item-based CF finds similarity between predicted item and sets of items that are similar to the items that a target user gives high rating in the past. While user-based CF finds similar users to a target user and gives him/her recommendations based on what other people with similar consumption patterns appreciated [6].

Since both content-based filtering and collaborative filtering have advantages and disadvantages, hybrid recommender system combines those two methods.

Recent studies employed generative adversarial networks (GANs) approach in recommendation system. GANs is a deep generative model which consists of a generator and a discriminator. Both are trained under the adversarial learning idea. The objective of GANs is to estimate the latent distribution of real data samples and generate new samples from that distribution. GAN has been extensively applied for many tasks [7], including recommendation system by extracting latent user and item characteristics from discriminator to calculate rating scores.

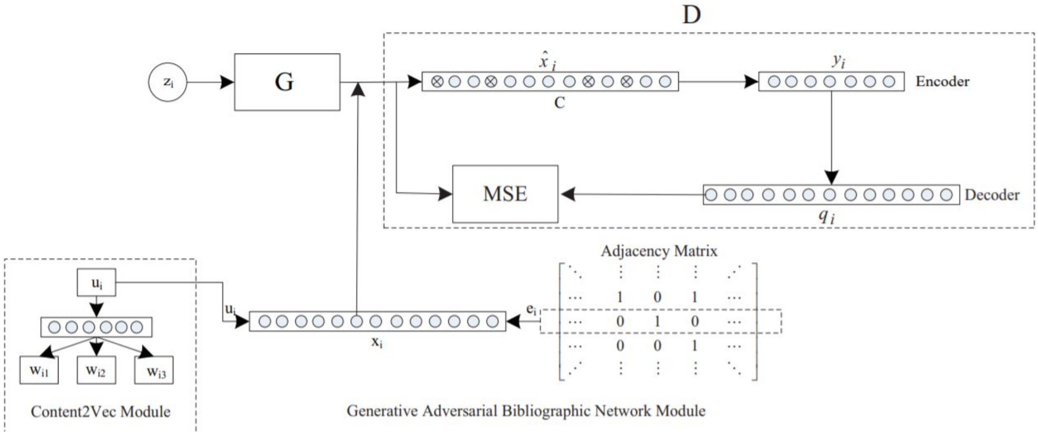

Figure 1 Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation (GAN-HBNR)

In recent work, which is GAN Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation (GAN-HBNR), they applies GANs into recommendation systems by using a denoising autoencoder (DAE) as a discriminator [8]. DAE is an autoencoder that randomly corrupts input to prevent overfit problem. [10] It can extract latent feature representation of the input. DAE is applied in GAN-HBNR by extracting document profile and author profile to calculate similarity score. GAN-HBNR (Figure 1) has two input vectors (xi) which are the output of context2vec of document (u_i) that represents document embedding, and relationship graph (Adjacency Matrix) of author and document (e_i) [5] , they concatenate two of them as input x1 and put it into discriminator (D). The generator (G) also creates data from random noises(zi) and put it into discriminator. The discriminator will try to discriminate between real and fake inputs. After training in several steps, They get a hidden feature representation vector of the document profile and author profile from an encoder. These vectors will be used to calculate similarity score by using equation 1, where r_q is similarity scores, V_{PR} is the vector representation of training documents, v_{qt} is the vector representation of the testing document, V_{AR} is the vector representation of authors related to training documents, v_{qa} is the vector representation of the authors related to testing document.

$$r_q = V_{PR} v_{q_t}^T + V_{AR} v_{q_a}^T \qquad (1)$$

In the GAN-HBNR model, I assume that document characteristics is user embedding, and relationship graph of author and document matrix is user-item matrix. This model is a content-based filtering because it has only a user embedding without using the opinions of other users or neighbors to predict the rating.

In this work, I will use CF method which considers the opinions of other users or neighbors to predict the rating. Since content based filtering approach uses only a personalized historical data, if I add other people's opinion, the model should be more accurate. By applying CF method in our work, I also add similarity matrix into the input. In case of user-based CF, I add user similarity matrix which is calculated from cosine similarity between a target user profile and other user profiles. While item-based CF, I add item similarity matrix which is calculated from cosine similarity between a target item profile and other item profiles. Furthermore, I will use a transformer instead of denoising autoencoder as discriminator. Transformer can be used to extract latent representation of the input like in DAE. However, its architecture is more complex than the DAE. Transformer is a type of neural network that transforms an input sequence to an output sequence by using an encoder and a decoder. It has attention-mechanisms that pay attention to the important items. Therefore, transformer can extract latent representation of the input better than DAE.

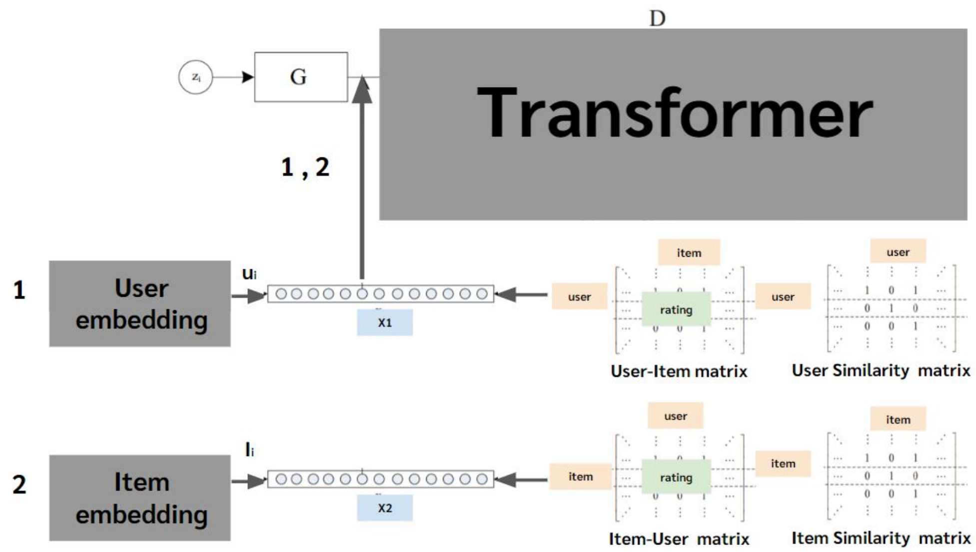

Figure 2 Proposed model : Incorporate Transformers with GANs for Item based and User based Collaborative filtering recommendation model.

In the proposed model (Figure 2) , I will use user embedding(ui) or item embedding(Ii), user-item rating matrix, and similarity matrix as the inputs. In the case of user-based CF, I will use a user similarity matrix. In case of item-based CF, I will use item similarity matrix. Then, I concatenate three of them as input x1 (user-based CF) and x2 (item-based CF) and put it into the discriminator. The generator also creates data from random noises and puts it into a discriminator. The discriminator will try to discriminate between real and fake inputs. After training in several steps, I get a hidden feature representation vector of the user profile in case of user-based CF and item profile in case of item-based CF from an encoder. These vectors will be used to calculate ratings in the collaborative filtering method. In case of user-based CF, the prediction state is calculated by using equation 2, where r_{ij} is predicted rating for target user i toward target item j, u_i is target user, u_k is neighbor user, r_{kj} is rating that neighbor has rated to item j. In case of item-based CF, the prediction state is calculated by using equation 3, where r_{up} is predicted rating for target user u toward target item p, i is another item that is similar to item p, and r_{ui} is rating that target user u rates for similar item i. After that, I calculate the final rating which is average of user-based CF rating score and item-based CF rating score and the result of recommendation will be obtained.

$$r_{ij} = \frac{\sum_k similarities(u_i,u_k)(r_{kj})}{number\ of\ ratings} \tag{2}$$

$$r_{up} = \frac{\sum_k similarities(i,p)(r_{ui})}{\sum_k similarities(i,p)} \tag{3}$$

To conclude, I propose a new recommendation system model to predict rating by using generative adversarial networks model on user preference, user similarities, and item similarities to perform user-based and item-based collaborative filtering and compare the efficiency between the proposed model and the previous recommendation models by using accuracy.

Objective

1. To propose a new recommendation system model to predict rating by using generative adversarial networks model on user preference, user similarities and item similarities to perform user-based and item-based collaborative filtering.
2. To compare the efficiency between the proposed model and the previous recommendation model by using accuracy.

Scope of work

1. I will use MovieLens 100K Dataset. There are 100,00 ratings in the range of 0 to 5 (max rating is 5.0 and min rating is 0.5) from 1000 users on 1700 movies including userId, movieId, rating, title and genres.
2. I compare the efficiency of the proposed model with the research work of Xiaoyan Cai, Junwei Han, Libin Yang on Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation

Procedure

1. Study research and academic articles related to recommendation systems.
2. Identify problem and disadvantage of previous work.
3. Design and analysis solutions to the problem of recommendation systems.
4. Implement and Improve the proposed model
5. Evaluate the proposed model.
6. Analyze and discuss the results in order to improve the proposed model in the future
7. Create related document such as report.

The aforementioned procedure can be written in Gantt Chart as follows

Procedure	2019					2020			
	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APL
1. Study research and academic articles related to recommendation systems.									
2. Identify problem and disadvantage of previous work.									
3. Design and analysis solutions to the problem of recommendation systems.									
4. Implement and Improve the proposed model									
5. Evaluate the proposed model.									
6.Analyze and discuss the results in order to improve the proposed model in the future									
7. Create related document such as report.									

Expected Benefits Gain

1. Benefits in terms of knowledge and experience for implementor.
   a. Get to learn the theory and operation of building new method for recommendation system.
   b. Get to apply knowledge in computer science to solve problems in the recommendation system.
2. Benefits in solving the social problem.
   a. To contribute to previous research in order to improve research in this field.
   b. To solve the problem of rating behavior of different users.
   c. To better improve the recommendation system.

Equipment and Tools

1. Hardware
    a. Computer with Windows 10 64-bit operating system 2.60 GHz Intel Core i7 RAM 16 GB
2. Software
    a. Google Collaboratory

Budget

1. Brother BT-6000BK black bottle refill	1	Bottle	250	Baht
2. Brother Refill Ink Brother BT-6000C / M /	3	Bottles	660	Baht
YA4	2	Reams	240	Baht
3. DoubleA	2	CDs	20	Baht
4. CD	1	Piece	1,670	Baht
5. WD HDD 1TB My Passport 2017	1	Piece	4,270	Baht
6. SSD 250 GB				
		Total	7,110	Baht

Note: Budget can be changed as appropriate and the price as mentioned above is referenced from http://www.officemate.co.th/ and https://www.jib.co.th

Reference

1. Rishabh Mall.Recommender System[internet].2019[cited 2018 Oct 20]Available from:https://towardsdatascience.com/recommender-system-a1e4595fc0f0
2. Zhang, Shuai et al. “Deep Learning Based Recommender System.” ACM Computing Surveys 52.1 (2019): 1–38. Crossref. Web.
3. Wikipedia contributors. (2019, October 21). Recommender system. In Wikipedia, The Free Encyclopedia. Retrieved 06:54, October 22, 2019, from https://en.wikipedia.org/wiki/Recommender_system
4. Abhinav Ajitsaria .Build a Recommendation Engine With Collaborative Filtering[internet].2018 [cited 2018 Oct 20] Available from:https://realpython.com /build-recommendation-engine- collaborative-filtering/
5. Guangxia Xu, Zhijing Tang, Chuang Ma, Yanbing Liu, and Mahmoud Daneshmand, “A Collaborative Filtering Recommendation Algorithm Based on User Confidence and Time Context,” Journal of Electrical and Computer Engineering, vol. 2019, Article ID 7070487, 12 pages, 2019. https://doi.org/10.1155/2019/7070487.

6. PETER, B., MELKER, F. Comparison of User Based andItem Based Collaborative Filtering.STOCKHOLM, SVERIGE 2017, 2017.
7. Wang, Kunfeng & Gou, Chao & Duan, Yanjie & Yilun, Lin & Zheng, Xinhu & Wang,
Fei-Yue. (2017). Generative Adversarial Networks: Introduction and Outlook. ACTA AUTOMATICA SINICA. 4. 588-598. 10.1109/JAS.2017.
8. Cai, Xiaoyan, Han, Junwei, AND Yang, Libin. "Generative Adversarial Network Based
Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation" AAAI Conference on Artificial Intelligence (2018): n. pag. Web. 22 Oct. 2019
9. Arden Dertat .Applied Deep Learning - Part 3: Autoencoders[internet].2017[cited 2018 Oct 20]Availablefrom:https://towardsdatascience.com/applied-deep -learning-part-3-autoencoders-1c083af4d798
10. Jannik Zürn .But what is an Autoencoder?[internet].2018[cited 2018 Oct 20] Available from:https://medium.com/@jannik.zuern/but-What-is-an -autoencoder-26ec3386a2af
11. Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. Recommender Systems Handbook.
Springer-Verlag New York, Inc., 2010.
12. Zhao, J.; Mathieu, M.; and Lecun, Y. 2016. Energy-based generative adversarial network. *arXiv: 1609.03126*.
13. Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David,Ozair,Sherjil,Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.
14. Wikipedia contributors. "Transformer (machine learning model)." Wikipedia, The Free
Encyclopedia. Wikipedia, The Free Encyclopedia, 24 Oct. 2019. Web. 24 Oct. 2019.

BIOGRAPHY

Miss Thunchanok Tangpong
Department of Mathematics and Computer Science
Faculty of Science, Chulalongkorn University
Email: ttp.pun@gmail.com
Interested Field: Recommender System, Machine Learning , Data Mining, Deep learning, Natural Language processing and Image processing