โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ ระบบแนะนำผู้ใช้แบบผสม (การกรองเนื้อหาและการกรองแบบร่วมมือกัน) โดยมีพื้นฐานอยู่บน BERT
Hybrid (Content-based filtering and Collaborative filtering) recommender system based on BERT

ชื่อนิสิต นางสาวชนาภา ชาญณรงค์ 593 36163 23
นางสาวชวิศา เผ่าศิริกุล 593 36186 23

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2562

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ระบบแนะนำผู้ใช้แบบผสม (การกรองเนื้อหาและการกรองแบบร่วมมือกัน) โดยมีพื้นฐานอยู่บน BERT

นางสาวชนาภา ชาญณรงค์
นางสาวชวิศา เผ่าศิริกุล

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Hybrid (Content-based filtering and Collaborative filtering) recommender system based on BERT

Ms. Chanapa Channarong
Ms. Chawisa Paosirikul

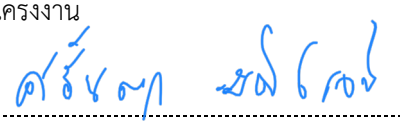
A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science Program in Computer Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2019

หัวข้อโครงงาน	ระบบแนะนำผู้ใช้แบบผสม (การกรองเนื้อหาและการกรองแบบร่วมมือกัน) โดยมีพื้นฐานอยู่บน BERT
โดย	นาวสาวชนาภา ชาญณรงค์
	นางสาวชวิศา เผ่าศิริกุล
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงงานหลัก	รองศาสตราจารย์ ดร.ศรันญา มณีโรจน์
อาจารย์ที่ปรึกษาโครงงานร่วม	ผู้ช่วยศาสตราจารย์ ดร.กิติพร พลายมาศ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

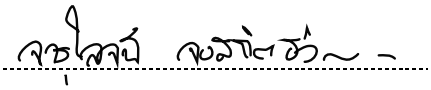
(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

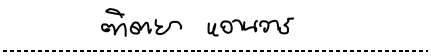


(รองศาสตราจารย์ ดร.ศรันญา มณีโรจน์) อาจารย์ที่ปรึกษาโครงงานหลัก

(ผู้ช่วยศาสตราจารย์ ดร.กิติพร พลายมาศ) อาจารย์ที่ปรึกษาโครงงานร่วม



(ผู้ช่วยศาสตราจารย์ ดร.จารุโลจน์ จงสถิตย์วัฒนา) กรรมการ



(ผู้ช่วยศาสตราจารย์ ดร.ฑิตยา หวานวารี) กรรมการ

นางสาวชนาภา ชาญณรงค์, นางสาวชวิศา เผ่าศิริกุล: ระบบแนะนำผู้ใช้แบบผสม (การกรองเนื้อหาและการกรองแบบร่วมมือกัน) โดยมีพื้นฐานอยู่บน BERT. Hybrid (Content-based filtering and Collaborative filtering) recommender system based on BERT อ.ที่ปรึกษาโครงงานหลัก : รศ.ดร.ศรันญา มณีโรจน์, อ.ที่ปรึกษาโครงงานร่วม : ผศ.ดร.กิติพร พลายมาศ, 56 หน้า.

ระบบแนะนำเป็นระบบที่แนะนำไอเท็มให้กับผู้ใช้โดยพิจารณาจากความชื่นชอบของผู้ใช้เป็นหลัก ซึ่งวิธีพื้นฐานที่ใช้ในระบบแนะนำมีอยู่สองวิธีได้แก่ การกรองเนื้อหาและการกรองแบบร่วมมือกัน แต่วิธีพื้นฐานดังกล่าวยังมีประสิทธิภาพในการแนะนำไม่เพียงพอ ดังนั้นจึงมีผู้คิดค้นวิธีอื่นๆ เช่น โครงข่ายประสาทเทียมแบบการกรองเนื้อหา และโครงข่ายประสาทเทียมแบบการกรองแบบร่วมมือกัน แต่อย่างไรก็ตามวิธีเหล่านี้ต่างก็ไม่ได้ให้ความสำคัญกับลำดับของการปฏิสัมพันธ์ของผู้ใช้เป้าหมาย ด้วยเหตุนี้ในปัจจุบันจึงมีการทำระบบแนะนำแบบลำดับเพิ่มมากขึ้น หนึ่งในนั้นก็คือ BERT4Rec ที่นำ BERT ที่เป็นโมเดลด้านความเข้าใจในภาษาเข้ามาทำระบบแนะนำแบบลำดับ ซึ่งในงานนี้สนใจแต่ประวัติที่เป็นลำดับของไอเท็มของผู้ใช้เป้าหมาย และไม่ได้สนใจข้อมูลของผู้ใช้คนอื่นๆในระบบเลย กล่าวคือมีเพียงวิธีการของการกรองเนื้อหา ดังนั้นผู้วิจัยจึงเสนอวิธีการใหม่ที่เรียกว่า ระบบแนะนำผู้ใช้แบบผสมโดยมีพื้นฐานอยู่บน BERT ซึ่งเป็นการนำ BERT เข้ามาทำทั้งในส่วนการกรองเนื้อหาและการกรองแบบร่วมมือกัน โดยในส่วนของการกรองเนื้อหาจะพิจารณาประวัติที่เป็นลำดับของไอเท็มของผู้ใช้เป้าหมายเช่นเดียวกันกับโมเดล BERT4Rec แต่จะให้ผลลัพธ์เป็น โปรไฟล์ของผู้ใช้เป้าหมาย และในส่วนของการกรองแบบร่วมมือกันจะนำผู้ใช้คนอื่นที่เคยให้คะแนนไอเท็มเป้าหมายมาทำเป็นลำดับและให้ผลลัพธ์เป็น โปรไฟล์ของไอเท็มเป้าหมาย จากนั้นนำผลลัพธ์ของการกรองเนื้อหาและการกรองแบบร่วมมือกันมาคำนวณคะแนนโดยใช้วิธีการของ NCF ผู้วิจัยได้ใช้ข้อมูลของ MovieLens-1M เพื่อเปรียบเทียบประสิทธิภาพของวิธีการที่ได้นำเสนอกับโมเดล BERT4Rec โดยจะเปรียบเทียบในด้านของความถูกต้องโดยใช้วิธีการของ NDCG ซึ่งจากผลการทดลองพบว่าวิธีการที่ผู้วิจัยเสนอให้ผลลัพธ์ที่ดีกว่าโมเดล BERT4Rec

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต .....

ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงานหลัก .....

ปีการศึกษา 2562 ลายมือชื่อ อ.ที่ปรึกษาโครงงานร่วม .....

5933616323, 5933618623: MAJOR COMPUTER SCIENCE
KEYWORDS : RECOMMENDER SYSTEM / BERT / HYBRID RECOMMENDATION / SEQUENTIAL RECOMMENDATION

CHANNAPA CHANNARONG, CHAWISA PAOSIRIKUL: HYBRID (CONTENT-BASED FILTERING AND COLLABORATIVE FILTERING) RECOMMENDER SYSTEM BASED ON BERT. ADVISOR : ASSOC. PROF. SARANYA MANEEROJ, Ph.D., CO-ADVISOR : ASST. PROF. KITIPORN PLAIMAS, Ph.D., 56 pp.

Recommender system (RS) is the system that recommends items to users based on user preference. There are two main methods that are used in RS, including content-based filtering (CBF) and collaborative filtering (CF). However, these two main methods are not effective enough to get the better recommendation. Hence, many people have explored other methods such as neural content-based filtering and neural collaborative filtering. But those methods do not focus on the interaction sequence of the target user. Recently, there are researches that proposed to model sequential recommendations based on users' historical interaction. One of these is BERT4Rec which takes BERT, a language understanding model, to model a sequential recommendation. This method only considers the historical sequence (item sequence) of the target user and does not consider the interaction of other users in the system that is content-based filtering approach. For this reason, we proposed a new method called Hybrid recommender system based on BERT, which applied BERT on both CBF and CF. For CBF, we consider the item sequence of the target user as same as BERT4Rec, but the result of our proposed method is the target user profile. For CF, we consider other users who used to interact with the target item and call it as user sequence of the target item. The result of the CF side is the target item profile. Finally, after obtaining the results on both CBF and CF side, we use it to predict the rating score by the NCF approach. To evaluate our proposed method, we compared it with BERT4Rec on MovieLens-1M dataset in terms of accuracy by the NDCG approach. The experimental result shows that our proposed method outperforms BERT4Rec.

Department : Mathematics and Computer Science Student's Signature Channapa Channarong
Student's Signature Chawisa Paosirikul
Field of Study : Computer Science Advisor's Signature Saranya Maneeroj
Academic Year : 2019 Co-advisor's Signature K. Plaimas

ACKNOWLEDGEMENTS

First of all, we would like to express our sincere thanks to our senior project advisor, Assc. Prof. Saranya Maneeroj, Ph.D. for dedication herself and encouraging us throughout this work. Without her help and all support that we received from her, we would not have achieved so far and this work would have never been accomplished. Moreover, she is not only teaching us academic knowledge but also many methodologies in life. In addition, we are grateful to our senior project co-advisor, Asst. Prof. Kitiporn Plaimas, Ph.D. for helping and supporting us with this work. She reviewed our report and grammatical checking in it.

We would like to express our gratitude to our committee, including Asst. Prof. Jaruloj Chongstitvatana, Ph.D. and Asst. Prof. Dittaya Wanvarie, Ph.D. for generously offering their time, guidance, and goodwill throughout our project and review of our project proposal. Furthermore, Asst. Prof. Dittaya Wanvarie, Ph.D. has advised us on how to use the Amazon web service for training our model.

We are also thankful to our family and our friends for all their supporting and encouraging us throughout the period of this work. For our family, they give us love, good wishes and always be by our side. For our friends, they always help when we face the problem in implementation and they are good friends and make us have good memories throughout university life.

Finally, we are thankful to the Department of Mathematics and Computer Science, Chulalongkorn University, for the property, research operations, and budget for the implementation of this research.

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I
INTRODUCTION

1.1 Background and rationale

A recommender system (RS) plays an essential role in our everyday life in which we have seen on many websites and applications like YouTube, Facebook, Netflix, and Amazon. Furthermore, various technologies were developed. It makes people be able to access much information that leads to information overload. Therefore, the recommendation system is an essential part of supporting user decisions.

Collaborative filtering is one of the primary methods commonly used in the recommender system. This method measures the similarity between a target user and its neighboring users, and calculates the rating score to create a list of recommended items for the target user. However, there is the cold start user problem when a new user enrolls in the system since this user will have incomplete information. In other words, he/she has never give ratings to items or give ratings only through a small number of items. This problem causes a poor-quality recommendation. This type of the recommender system also causes the sparse data. Mostly, since the number of items is greater than the users so it makes the user not be able to rate the item enough to calculate similarities between users and find the neighbor of a target user. For this reason, this method requires a large amount of data to calculate the recommendation.

Another primary method is content-based filtering, which measures the similarities between an item that is going to be recommended for a target user, and items that the target user was familiar with in the past by considering the content that describes the items. The advantage of this method is that it does not require much data for processing or there is no cold start. But the limitation is that the set of items that were recommended for a target user is quite specific because it cannot recommend new items that the target user has never seen before.

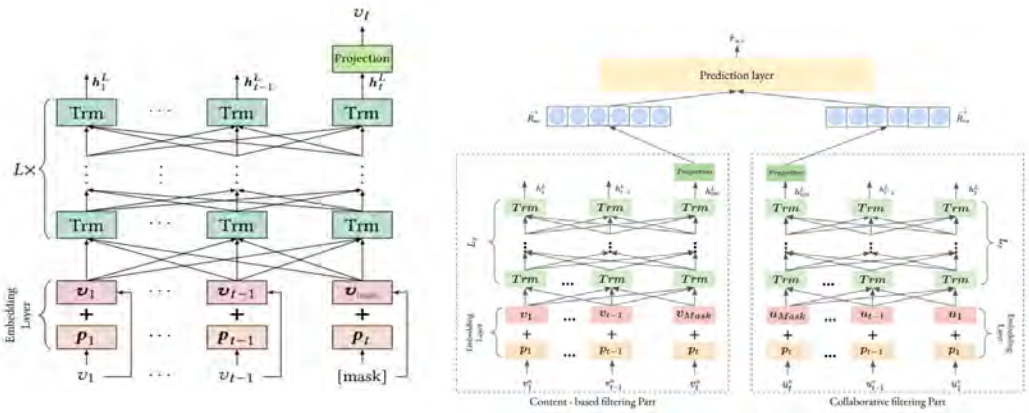
However, the two methods above have both advantages and disadvantages. Therefore, they were combined to improve the recommender system to be more efficient which still preserves the advantages and eliminates some disadvantages — this is called hybrid recommender system approach.

BERT stands for Bidirectional Encoder Representations from Transformers. It is a model that brings the encoder parts of Transformer [13] to generate a language model. The regular Transformer is trained left-to-right to predict the next word (target word) of each position in the input sequence. But the regular Transformer has limitations to train in bidirectional because joint context on both left and right would cause information leakage which allows each word to see the target word hence the model would not learn anything useful. To alleviate this problem, BERT applied bi-direction training to the regular Transformer by adopting masked language modeling, which randomly masks words and assigns the model to predict that word based on their surrounding context.

The earlier work [9] adopts the deep bidirectional self-attention model BERT to the sequential recommender system for predicting the next item that users are likely to interact with. By giving the historical interaction and applying masked language modeling [12] randomly on the items, then predicted the masked items based on the surrounding items. The previous model extracts the user profile by considering the similarity on every pair of items in his/her historical item sequences. When the new item has entered, this model will predict which item should be the next (target item). Moreover, their model only uses historical data of a target user without considering interaction information of other users toward a target item which is the content-based filtering approach. It will be better if collaborative filtering can be applied in BERT for making their model have higher accuracy that is the Hybrid recommendation approach. By looking at the information of other users that interact with a target item. Instead of an attention on item sequences of the target user, we are interested in user sequences. Users in this sequence are the users who used to rate or interact with the target item. In order to extract which users affect the target item and would be neighbors of the target user, we will apply the attention mechanism in BERT by feeding another input which is the user sequences of the target item.

In this work, we propose a Hybrid (Content-based filtering and Collaborative filtering) recommender system that applying the BERT model. In addition to content-based in the previous model, we intend to add user-based collaborative filtering in order to consider how other users rate the target item. From the previous model in Figure 1.1 (a), it shows that it has employed BERT on only item sequences rated by the target user that is the content-based filtering approach, while our model in Figure 1.1 (b)

incorporates a collaborative filtering approach by feeding another input sequence including all users who used to rate the target item into the model. After taking input as an item sequence (content-based filtering part) and a user sequence (collaborative filtering part), we will obtain target user profile and target item profile respectively. For target user profile, it consists of the similarity on every pair of items on the historical sequence of the target user. Consequently, it provides the information which shows whether the next item is the target item. Meanwhile, a target item profile consists of the similarity on every pair of users which shows who are the neighbors of the target user. In the prediction state, we apply NCF [5] to predict the rating score between target user profile and target item profile.



(a) BERT4Rec model architecture (b) HybridBERT4Rec model architecture

Figure 1.1 BERT4Rec model and HybridBERT4Rec model architecture

1.2 Objectives

1. To propose a new method that applied BERT on both content-based filtering approach and collaborative filtering approach.
2. To compare performance in terms of accuracy between proposed model and traditional model (BERT4Rec [9]).

1.3 Scope

1. Use the MovieLens-1M [4] dataset that consist of 6,040 users, 3,706 movies and 1,000,209 ratings. The lowest rating score is 1 and the highest rating score is 5.

2. Compare performance in terms of accuracy of the proposed approach with the research of Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang about BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer.

1.4 Project Activities

1. Outline of study

   1.1 Study the architecture of the current recommendation system.

   1.2 Study the research and academic articles in recommendation systems.

   1.3 State the pros and cons of the previous methods.

   1.4 Analyze and design methods to solve the problem.

   1.5 Develop a correctness test of the proposed method.

   1.6 Perform an experiment to measure the performance of the proposed system.

   1.7 Analyze and discuss the experimental results.

   1.8 Summarize and write a report.

2. Timeline of study

   From the outline of study above, we can write the Gantt chart as below.

Table 1.1 Gantt chart of project activities

Procedures	2019						2020			
	07	08	09	10	11	12	01	02	03	04
1. Study the architecture of the current recommendation system.										
2. Study the research and academic articles in recommendation systems.										
3. State the pros and cons of the previous methods.										
4. Analyze and design methods to solve the problem.										

5. Develop a correctness test of the proposed method.										
6. Perform an experiment to measure the performance of the proposed system.										
7. Analyze and discuss the experimental results.										
8. Summarize and write a report.										

1.5 Benefits

1. For Researcher

   1.1 Learn the operation and algorithm of recommender systems.

   1.2 Learn the theory and practice in creating a new approach of recommender systems.

   1.3 Apply knowledge to create a recommender system.

   1.4 Practice the skill of work planning

   1.5 Practice in solving problems that occur during the operation.

2. For Business and Society

   2.1 Develop knowledge that will be applied to a business in the future.

   2.2 Continue to develop a new knowledge which is beneficial to the research industry.

1.6 Report Outlines

The rest of this report is organized as followings:

1. Chapter 2 Literature Review: will present the related research, related knowledge, and the related evaluation metrics in the recommender system.
2. Chapter 3 Methodology: will explain the proposed method and its process in detail.
3. Chapter 4 Results: will explain the dataset and evaluation metric that used in this experiment and the experiment results of proposed method.
4. Chapter 5 Conclusion: will present the conclusion and future plan of this work.

CHAPTER II
LITERATURE REVIEW

In this chapter, the recommender system was introduced which consists of collaborative filtering and content-based filtering. Next, the neural collaborative filtering and neural content-based filtering are also introduced. To consider the order in users' historical interaction, the sequential recommendation was explained. Before applying BERT to the sequential recommendation, Transformer and BERT are first introduced. After that, BERT4Rec was explained which is the model that applied BERT in the sequential recommendation task. Finally, the negative log-likelihood and l2-normalization were explained in detail for the loss function.

2.1 Recommender System

Recommender system is the system that recommends items to the target user by looking at the users' preference or the users' rating toward the item. Because of the information overload problem, RS was introduced and there is much research on RS for getting a better recommendation. This will help the user easily to access the information that the user was interested in and help the user make a decision. Nowadays, many websites and applications like YouTube, Facebook, Netflix, and Amazon all use RS on it. There are two main general methods in RS which are collaborative filtering and content-based filtering.

2.1.1 Collaborative Filtering

Collaborative filtering (CF) is the method that recommends an item by looking at the rated item set of the neighbor user who has the same preference pattern to the target user. This method can alleviate the serendipitous problem [1] which is the items that were recommended to target users are quite specific and target users unable to receive the new style of items. For example, if *userA* has rated only superhero movies. *UserA* will receive only superhero movies and cannot receive the romantic movie that *userA* is also interested in. The details of how CF works are as follows: First, CF receives the input as a user-item rating matrix as shown in Figure 2.1.

	item 1	item 2	item 3	item 4
user 1	-	5	3	4
user 2	2	-	1	1
user 3	1	4	5	-
user 4	5	3	-	3
user 5	2	-	4	3

Figure 2.1 User-Item rating matrix

From Figure 2.1, suppose that the target user is *user3*. To predict the rating score of *user3* toward *item4* ($\hat{r}_{ui}$), CF has to find who is the neighbor of *user3* by computing the cosine similarity between *user3* and others on co-rated items using Equation (1),

$$cosine(u, v) = \frac{\sum_{i=1}^{n} r_{ui} r_{vi}}{\sqrt{\sum_{i=1}^{n} r_{ui}^2} \sqrt{\sum_{i=1}^{n} r_{vi}^2}} \quad (1)$$

where u denotes the target user. v denotes another user. i denotes the target co-rated items. n denotes the number of co-rated items. r_{ui} denotes the rating that user u rate on item i. And r_{vi} denotes the rating that user v rate on item i. Therefore, the cosine similarity score between *user3* and others are shown in Figure 2.2.

$$cosine(user3, user1) = \frac{4(5) + 5(3)}{\sqrt{4^2 + 5^2} \times \sqrt{5^2 + 3^2}} = 0.94$$

$$cosine(user3, user2) = \frac{1(2) + 5(1)}{\sqrt{1^2 + 5^2} \times \sqrt{2^2 + 1^2}} = 0.61$$

$$cosine(user3, user4) = \frac{1(5) + 4(3)}{\sqrt{1^2 + 4^2} \times \sqrt{5^2 + 3^2}} = 0.71$$

$$cosine(user3, user5) = \frac{1(2) + 5(4)}{\sqrt{1^2 + 5^2} \times \sqrt{2^2 + 4^2}} = 0.96$$

Figure 2.2 Cosine similarity score between *user3* and others

After obtaining the cosine similarity score, CF will select the top-N neighbors of the target user by considering the top-N highest cosine similarity score between the target user and others. For example, CF will select the top-2 neighbors of

target *user3* that are *user5* and *user1* respectively. In the rating prediction step, given the top-N neighbors information, CF calculates the rating score by using Equation (2) as follows:

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} sim_{uv} r_{vi}}{\sum_{v \in N(u)} sim_{uv}} \quad (2)$$

where $N(u)$ denotes all neighbors of the target user u and sim_{uv} denotes the cosine similarity between user u and user v. Thus, the rating score of *user3* toward *item4* is 3.49 as shown in Figure 2.3.

$$\hat{r}_{u3i4} = \frac{0.96(3) + 0.94(4)}{0.96 + 0.94} = 3.49$$

Figure 2.3 Rating score of *user3* toward *item4*

Although CF can solve the serendipitous problem but there is a limitation of CF called data sparsity because the number of items is larger than the number of users and users usually rate only through the small item. This makes CF difficult to find the neighbor of the target user.

2.1.2 Content-based Filtering

Content-based filtering (CBF) is the method that recommends the item which is similar to the items that the target user was familiar with in the past by looking at the information that describes the items. This method does not require any information of other users in the system since it considers only the data of the target user to make a prediction. The details of how CBF works are as follows: First, CBF extracts the users' preference or user profile and represents it in vector by analyzing the rated item set of the target user. Meanwhile, CBF also extracts the new item representation in vector by looking at the features that describe the item. Next, CBF will find the similarity between the user profile and other items. Finally, CBF creates a list of the top-N items that have the highest similarity score and recommends the item to

the target user. Although, the limitation of CBF is the serendipitous problem that the target users cannot receive the new style of items.

However, CF and CBF have both advantages and disadvantages. Consequently, using only the traditional approach like CF and CBF is not effective enough to get the better recommendation. Therefore, there is more research on the recommender system to find an effective method for recommendation.

2.2 Neural Collaborative Filtering

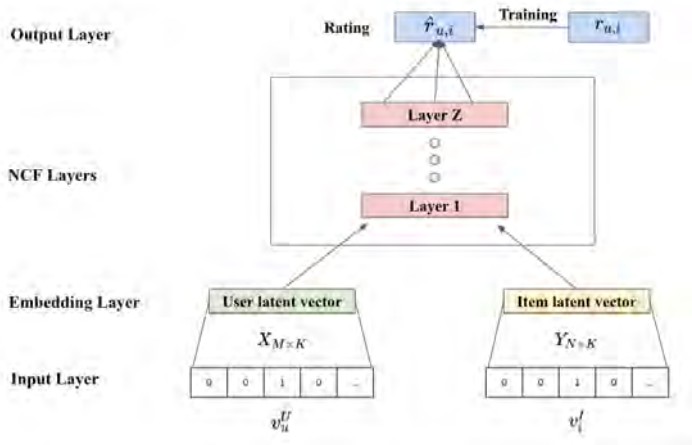


Figure 2.4 Neural collaborative filtering framework

Collaborative filtering approaches are famous approaches that researchers use on the recommender system. It is used for recommending the item based on users' historical preference. Matrix Factorization (MF) [6] is the most popular collaborative filtering approach, which predicts the rating by factoring the user-item rating matrix and represents user and item as latent features vector. Then, calculate the inner product between user and item latent vector to learn the users' preference on the item. MF has limitation which is the inability to learn the complex user-item interaction in a large number of latent factors K because it may face the overfitting problem.

The Neural Collaborative Filtering (NCF) [5] is another collaborative filtering approach, which replaces the inner product with Multi-Layer Perceptions (MLP) to model the users' preference. From Figure 2.4, let U be the set of users and V be the set of items. And let M and N be the number of users and items, respectively. The input layer consists of v_u^U and v_i^I which are the feature vectors with one-hot encoding to describe the user u and item i, respectively. Since those vectors are sparse, they are fed

to the embedding layer to be transformed into the dense vectors. The user embedding and item embedding can be assumed as the latent vector for user and item respectively. Then, feeding user embedding and item embedding to a neural collaborative filtering layer which comprises a multi-layer neural network to map the latent vectors and predict the rating score. The rating score of user u for item i ($\hat{r}_{u,i}$) is calculated by

$$\hat{r}_{u,i} = f(X^T v_u^U, Y^T v_i^I \mid X, Y, \theta_f) \tag{3}$$

where $X \in \mathbb{R}^{M \times K}$ and $Y \in \mathbb{R}^{N \times K}$, denotes the K latent factor matrix for users and items respectively. v_u^U and v_i^I is the vector with one-hot encoding for identifying a user u and item i. The function f is a multi-layer neural network that can be formulated as Equation (4) and θ_f is the model parameter for function f.

$$f(X^T v_u^U, Y^T v_i^I \mid X, Y, \theta_f) = \phi_{out}(\phi_Z(\dots \phi_2\left(\phi_1(X^T v_u^U, Y^T v_i^I)\right)\dots)) \tag{4}$$

ϕ_{out} and ϕ_Z are the mapping function for the output layer and the Z-th neural collaborative filtering (CF) layer. The model parameters are learned by minimizing the square loss function which represented as

$$L = \sum w_{u,i}(r_{u,i} - \hat{r}_{u,i})^2 \tag{5}$$

where $r_{u,i}$ is the true rating score of user u for item i and $w_{u,i}$ is the weight of training instances (u, i).

However, the collaborative filtering approach recommends the items for the target user by considering other user information. But it does not capture the relation among items in the users' historical sequence.

2.3 Neural Content-based Filtering

To alleviate the problem of capturing the relation between items in the users' historical sequence, the content-based filtering approach aims to learn the relation of items by considering the content that describes the items. Neural networks have become very trendy in recent years and have more strength than traditional content-based

filtering approaches. The four main factors that make neural-based approaches perform better are modeling in the non-linear interactions in the data with non-linear activations, possession of high flexibility, learning the underlying explanatory factors, and outperformance for sequential modeling tasks. Thus, many neural approaches are introduced and applied to the content-based scenario. For example, Convolutional Neural Network (CNN) generates local features and a pooling layer for concise representation. It is potent in processing unstructured multi-media data. The example of research that applies CNN in content-based is Deep Content-Based Music Recommendation [7]. It uses CNN to extract features from music signals. The convolutional kernels and pooling layers allow operations at multiple timescales. This content-based model can alleviate the cold-start problem of music recommendation. Multi-Layer Perceptron (MLP) use non-linear activation function and backpropagation for training, and Recurrent Neural Networks (RNN) use for learning sequential data.

2.4 Sequential Recommendation

Recently, many methods have been proposed to model sequential recommendations since most of the previous recommendation systems do not consider the order in users' historical interaction, which makes the recommendation systems not practical enough. Because of the next users' behavior would depend on the users' current interests. Thus, the sequential recommendation was proposed and became famous. Most of the sequential recommendations are based on Recurrent Neural Network (RNN) and its variants, Long Short-Term Memory (LSTM). The essential idea of those methods is proposing representation by encoding users' historical records into vectors in different ways.

2.4.1 Dynamic REcurrent bAsket Model (DREAM)

The model goal is to recommend a list of items that a user may purchase. DREAM [15] model is a next basket recommendation which baskets are the set of items that each user purchases. DREAM applied RNN by feeding baskets' representation from users' historical records to learn a representation of a users' interests at different times and called it the user's *dynamic representation*. The basket representation is generated by aggregating representation vectors of items in the basket. They use two

kinds of aggregation functions, which are *max-pooling* and *average-pooling*. For *max-pooling* basket representation $b_{t_t}^u$ can formulated as:

$$b_{t_t,k}^u = max(n_{t_i,1,k}^u, n_{t_i,2,k}^u, \dots) \quad (6)$$

where $b_{t_i,k}^u$ is k-th dimension basket vector representation of user u at times t_i and $n_{t_i,j,k}^u$ is the value of k-th dimension of the vector representation of the j-th item in basket $B_{t_i}^u$ and for *average-pooling* basket representation $b_{t_t}^u$ can formulated as:

$$b_{t_t}^u = \frac{1}{|B_{t_i}^u|} \sum_{j=1}^{|B_{t_i}^u|} n_{t_i,j}^u \quad (7)$$

In order to calculate the rating score for each user toward all items at each time step, they multiply the user's *dynamic representation* and the *item matrix* which is a stack of item embedding. A higher score indicates that the user is more likely to purchase the corresponding item.

2.4.2 The Multi-View Recurrent Neural Network (MV-RNN)

Instead of using only indirectly observable representation to represent the items like DREAM, MV-RNN [2] combines indirectly observable representation and directly observables (images and text description) representation to represent items. They represent users' historical records as:

$$I_u = (i_{u_1}, \dots, i_{u_{|I_u|}}) \quad (8)$$

where $i_{u_1}, \dots, i_{u_{|I_u|}}$ is items that the user u has purchased in chronological order and for each item content of image and text description. For items consist of two multi-view features: indirectly observable view and directly observable view. Indirectly observable view is latent feature of an item which represent as item embedding r_{u_t} defined by a vector:

$$r_{u_t} = i_{u_t} \qquad i_{u_t} \in \mathbb{R}^d \quad (9)$$

where i_{u_t} is the item of user u at timestep t.

Another view directly observable is multi-modal features consist of visual and textual features (f and g). They are obtained by GoogLeNet [10] and GloVe [8] weighted by TF-IDF respectively. In order to transform the original high-dimensional features to embedded low-dimensional visual and textual features (i_f and i_g), they learn two linear embedding matrices E and V:

$$i_f = Ef, \qquad i_f \in \mathbb{R}^d \tag{10}$$

$$i_g = Vg, \qquad i_g \in \mathbb{R}^d \tag{11}$$

There are three combinations of multi-view features:

1. Feature concatenation: the item representation is $i = [i_x; i_f; i_g]$ i is 3d-dimensional vector
2. Feature Fusion: the item representation is $i = [i_x; i_m]$

$$\text{where} \qquad i_m = i_f + i_g\,, \qquad i_m \in \mathbb{R}^d \tag{12}$$

3. Multi-Modal Marginalized Denoising AutoEncoder (3mDAE): a new fusion method to combine the multi-modal information to learn fusion features. This method can learn more robust features and tackle the lacking modalities problem. It base on the mDAE model The encoding process is represented by (10) and (11), and the corresponding hidden layer is built by (12). In the decoding process, we need to reconstruct the multi-modal input features. The mapping matrix in decoding process is just the transpose of the mapping matrix in encoding process

$$\hat{f} = E^T i_m$$
$$\hat{g} = V^T i_m \tag{13}$$

The MV-RNN model adopts the recurrent structure to capture dynamic changes in user's interest by feeding item representation into the LSTM layer.

2.4.3 Convolutional Sequence Embedding Recommendation Model (Caser)

Caser [11] incorporates the Convolutional Neural Network (CNN) and Latent Factor Model (LFM) to learn sequential features and user-specific features, respectively. It comprises three components: Embedding Look-up, Convolution Layer, Fully-connected Layer. For training CNN, they separate users' historical sequence S^u into two part which are L input items and T target items. To generates a training instance for user u is done by sliding a windows $L + T$ over user sequence. Embedding Look-up is a previous L items feature sequence in the latent space for user u at time step t can be formulated as:

$$E^{(u,t)} = \begin{bmatrix} Q_{S^u_{t-L}} \\ \vdots \\ Q_{S^u_{t-2}} \\ Q_{S^u_{t-1}} \end{bmatrix} \tag{14}$$

where Q is item embedding matrix.

In addition, they also have embedding P_u for a user u, representing user features in latent space. Convolution Layer has two types which are horizontal and vertical. Horizontal Convolutional Layer result is the vector o. It captures union-level patterns with multiple union sizes. Vertical Convolutional Layers are capturing point-level sequential patterns through weighted sums over previous items' latent representations. The result is the vector õ. Fully-connected Layer is performed to get more high-level and abstract feature they concatenate the outputs of the two convolutional layers and feed them into a fully-connected neural network layer:

$$z=\phi_a(W\begin{bmatrix} o \\ \tilde{o} \end{bmatrix} + b) \tag{15}$$

where W is the concatenation layer weight matrix, b is bias term and $\phi_a(\cdot)$ is activation function.

Finally, another fully-connected layer is added to estimate the probability of how likely a user wants to interact with each item at each time step. It can formulated as

$$y^{(u,t)} = W' \begin{bmatrix} z \\ P_u \end{bmatrix} + b' \tag{16}$$

where W' and b' are the weight matrix for the output layer and the bias term.

2.5 Transformer

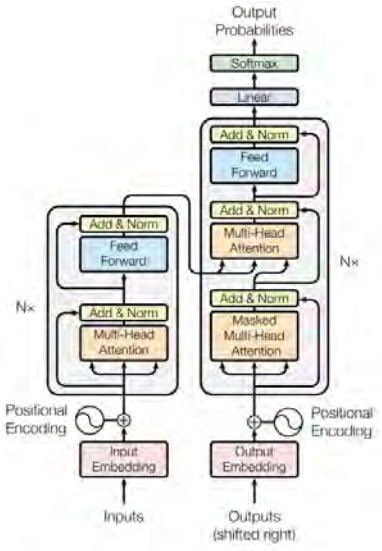


Figure 2.5 Transformer model architecture.
(Vaswani et al, 2017, p.3)

In the sequence task, Transformer [13] is a well known model. In general, Transformer is a model that was used in the natural language processing field, such as language modeling and machine translation. Transformer consists of two parts, which are an encoder and a decoder. Both parts aare composed of a stack of identical layers, whose main components are as follows:

1. Multi-Head Self-Attention Mechanism
2. Position-Wise Feed Forward Neural Networks

The significant method of the model is self-attention that makes Transformer be able to replace RNN and CNN. Self-attention is an attention mechanism relating to different positions of a single sequence in order to compute a representation of the sequence. In the Transformer, they adopt scaled dot-product attention and stack them in parallel that is called Multi-Heads Attention. The attention layer output is a weighted

sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (17)$$

where Q, K, V is matrix of query q, keys k, values v and d_k is dimension of vector k.

Since from their experiment computing multiple weighted sums is better representation rather than computing single attention (weighted sum of values), they compute multiple attention weighted sums. Each of these "Multiple-Heads" is a linear transformation of the input representation can formulated as:

$$head_i = Attention(QW^Q, KW^K, VW^V) \quad (18)$$

they compute Multi-Heads Attention as

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (19)$$

where h is number of head.

This provides the model to capture different aspects of the input and improve its expressive ability. Another module that is important in the Transformer is Positional Encoding. It works for the model to make use of the order of the sequence. They inject some information about the relative or absolute position of the tokens in the sequence to the embeddings. This is two sinusoids (sine, cosine functions) of different frequencies are used:

$$PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}}) \quad (20)$$

$$PE(pos, 2i + 1) = cos(pos/10000^{2i/d_{model}}) \quad (21)$$

where *pos* is the position of the token and *i* is the dimension.

An 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 feedforward neural network. They employ a residual connection

around each of the two sub-layers, followed by layer normalization, as shown in Figure 2.5. The residual connections are for retaining the position-related information which we are adding to the input representation/embedding across the network. The output of each sub-layer is

$$LayerNorm(x + Sublayer(x)) \tag{22}$$

where $Sublayer(x)$ is is the function implemented by the sub-layer itself and all sub-layers in the model, as well as the embedding layers, produce outputs of dimension d_{model} = 512.

Also, a decoder is composed of a stack of N = 6 identical layers which are similar as an encoder but insert a third sub-layer, which performs multi-head attention over the output of the encoder stack.

2.6 BERT

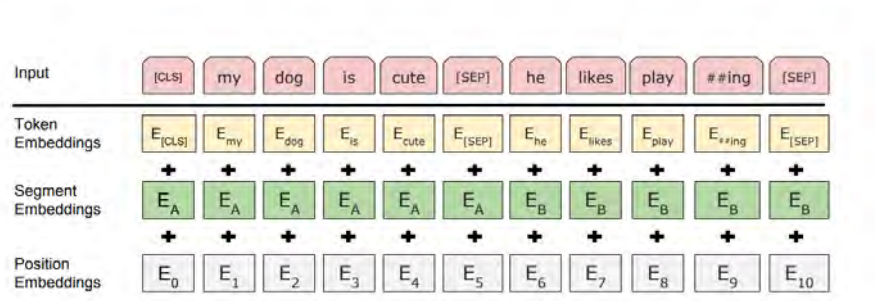


Figure 2.6 BERT input representation
(Devlin et al, 2018, p.5)

In addition to Transformers, BERT [3] is one of the successful methods in text understanding which achieves state-of-the-art results on text sequence modeling. BERT stands for Bidirectional Encoder Representations from Transformers. It developed from Transformer by adopting only an encoder to generate a language representation. Beside, pre-train BERT can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks. The regular Transformer model has limitations to train in bidirectional. Therefore, BERT applied bi-direction training to the

regular Transformer model for representations from an unlabeled text by jointly conditioning on both left and right context in all layers. However, language modeling is trained left-to-right by predicting the next word of each position in the input sequence. Jointly context on both left and right in bi-direction training would cause information leakage, which is allowing each word to see the target word so the model would not learn anything useful. BERT adopts masked language modeling [12], which randomly masks words. They mask 15% of all WordPiece tokens in each sequence at random. The training data generator chooses 15% of the token positions at random for prediction. They mask token in three different ways (1) replace with a special token [MASK] 80% of time, (2) replace with random token 10% of time, (3) replace with token itself 10% of time. To make BERT handle a single sentence and a pair of sentences tasks, an input representation is able to unambiguously represent in one token sequence. Thus, an input of BERT is sum of three types of embedding which are token embeddings, the segmentation embeddings and the position embeddings as shown in Figure 2.6. For token embeddings, they use WordPiece embedding [14] with a 30,000 token vocabulary. The first token of every sequence is a special token : [CLS]. The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks. For a pair of sentences, they differentiate the sentences in two ways. First, we separate them with a special token ([SEP]). Second, we add a learned embedding to every token indicating whether it belongs to sentence A or sentence B. Then, they feed input representation into BERT to reconstruct masked tokens to corresponding vocabulary based on their surrounding context. In order to predict words, they feed the final hidden vectors corresponding to the mask tokens into an output softmax over the vocabulary.

2.7 BERT4Rec

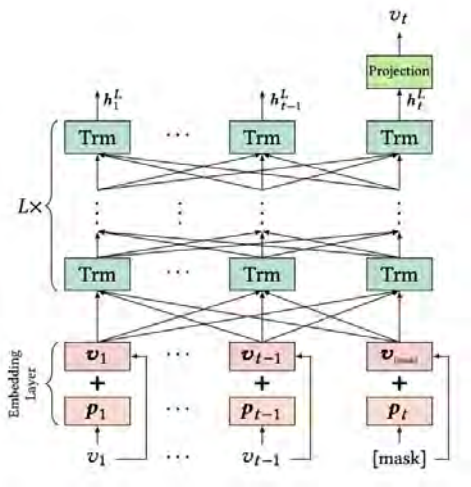


Figure 2.7 BERT4Rec model architecture
(Sun et al, 2019, p.3)

By the success of BERT, they have adopted BERT to model the sequential recommendation called BERT4Rec [9] model as shown in Figure 2.7. To predict what is the next item that a user is likely to interact with, let $V = \{v_1, v_2, \ldots, v_{|V|}\}$ be the items set. Firstly, the input of this model is the user's historical sequence or the item sequence which is the set of item that the user u used to interact with in the past in chronological order $S_u = \{v_1, v_2, \ldots, v_t\}$ where v_t is the item that the user u has interact at time step t. Next, they randomly mask the item sequence and replace it with a special token "[mask]". For example:

$$Input\ (item\ sequence): [v_1, v_2, v_3, v_4, v_5] \rightarrow\ [[mask_1], v_2, v_3, [mask_2], v_5]$$
$$Label:\ [mask_1] =\ v_1, [mask_2] =\ v_4$$

After randomly mask the input sequence, every token in item sequence are fed into the embedding layer which consists of input embedding and position embedding to extract the representation of each token v_t. After obtaining the item sequence embedding, they fed it into BERT model which has L transformer layer to learn the similarity on every pair of items in the sequence and define the loss function for each mask token as negative log-likelihood as follows in Equation (23):

$$L = \frac{1}{|S_u^m|}\sum_{v_m \in S_u^m} - logP(v_m = v_m^t | S_u') \quad (23)$$

where S_u' is the masked version of user u historical sequence, S_u^m is the randomly masked items in S_u, v_m^t is the true item for the masked item v_m, and P(.) is the probability that was defined in Equation (24).

$$P(v_t) = softmax(GELU(h_t^L W^p + b^p)E^V + b^O) \quad (24)$$

where W^p is the learnable projection matrix, b^p and b^O are bias terms, and E^V is the embedding matrix for the items set V.

Finally, this model produces the final hidden layer output of last token as h_t^L that shows the similarity of items on the users' historical sequence. The item that has the highest similarity, will be recommended to the user.

2.8 Negative log-likelihood

Negative log-likelihood (NLL) is cost function that is used as loss for machine learning models. The equation is as follows:

$$L(y) = -log\ (y) \quad (25)$$

where y is summed for all correct classes.

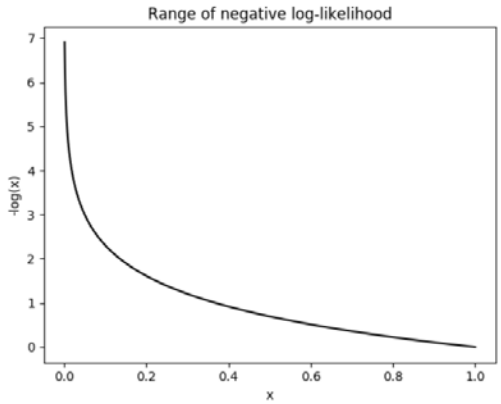


Figure 2.8.1 Range of negative log-likelihood graph

The range value of negative log-likelihood reaches zero to reach infinity which it reaches infinity when input is zero, and reaches zero when input is one. The details of how Negative log-likelihood works are as follows: First we have softmax output.

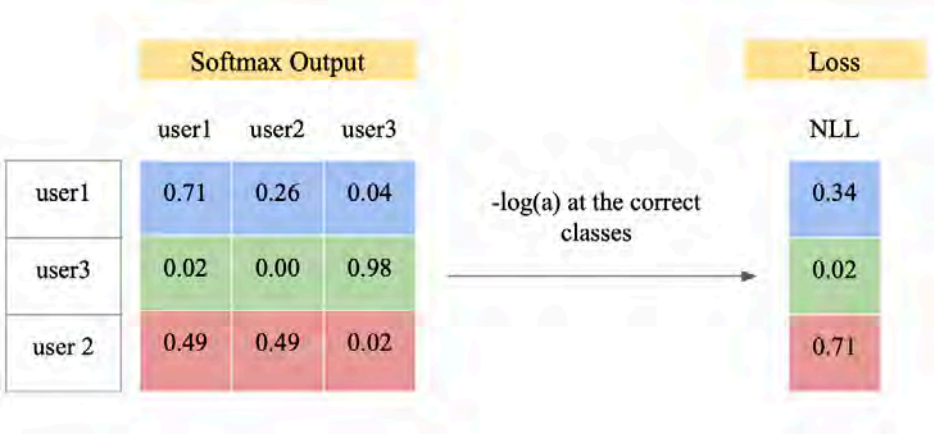


Figure 2.8.2 Example of negative log-likelihood calculation

Then, take negative log at the correct classes as shown in Figure 2.8.2. The better the prediction the lower the NLL loss.

2.9 L2-Normalization

L2-Normalization (also called l^2-norm) is one of the method that calculated the length of the vector. It is also a regularization method to prevent model overfitting. l^2-norm performs scale data input by maps vector values to values in [0,∞). The equation is as follows:

$$\| \bar{x} \| = \sqrt{\sum_i^n (x_i)^2} = \sqrt{(x_1)^2 + (x_2)^2 + \cdots + (x_n)^2} \tag{26}$$

where n is the number of element in vector $\bar{x}$.

L2-Normalization takes outliers in consideration during training. A linear regression model that implements. l^2-norm for regularization is called ridge regression. Regularization term is included when we calculate loss function. For l^2-norm, loss function can formulated as:

$$Loss = ERROR(y, \hat{y}) + \lambda \sum_{i=1}^{N} {x_i}^2 \tag{27}$$

From the above-related work, BERT4Rec is the model that can bring BERT, which is the current trend in text understanding, to the sequential recommendation. However, the BERT4Rec only uses the historical information of the target user and does not consider the information of other users to predict the next item that is the content-based filtering approach. In addition, the item that was recommended to the target user was specific because the target user will not receive the new style of item that the target user never seen before. It will be better if BERT can consider the information of other users for making more effective recommendations.

CHAPTER III
METHODOLOGY

BERT is the successful method for text understanding which considered both left and right context in the sequence of sentences. Consequently, there is the research that brings BERT to model the sequential recommendation in order to find the relation and representation of items in the users' historical sequence that is BERT4Rec model [9]. But this model only considers the historical data of the target user and has the serendipitous problem.

To solve the serendipitous problem and allow BERT to consider the interaction of other users in the system for making the recommendation to have more effective. We propose a new method that applies BERT on both content-based filtering and collaborative filtering called Hybrid (Content-based filtering and Collaborative filtering) recommender system based on BERT model which comprises three main parts: content-based filtering, collaborative filtering and prediction stage.

3.1 Content-based Filtering

In the content-based filtering side, our model aims to predict the next item that users likely to interact with by measuring the similarity between the historical item of the target user and the new item that the target user has never seen before. BERT4Rec is one of the models that can find the similarity among the items in the users' historical sequence. Therefore, we applied the BERT4Rec model in our model and the details are as follows.

Firstly, let $V = \{v_1, v_2, \ldots, v_{|V|}\}$ be the items set, then let the target user be *userA* and $S_A = \{v_1, v_2, \ldots, v_t\}$ be *userA*'s historical sequence or the item sequence where v_t is the item that *userA* used to rate at time step t. To make our model can consider both left and right context in the sequence, we aim to applied mask languages modeling [12] in our model which randomly masks 15% of all items in the sequence then replace it with special token "[mask]" as shown in Figure 3.1 and define as $S_A^m = \{v_1, v_m, v_3, \ldots, v_t\}$ where v_m is the item that was replaced with the token "[mask]". Then, the masked item will be store in set M_A. From this figure, the masked item is v_2.

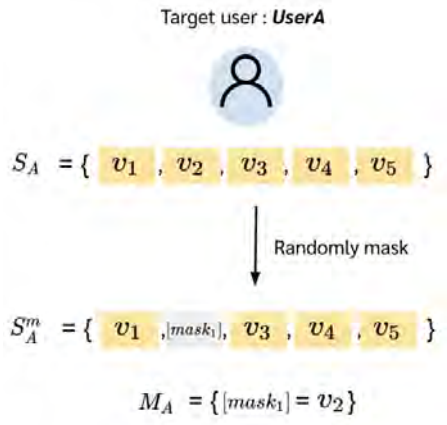


Figure 3.1 The input of content-based filtering part

After randomly mask the item in the item sequence, we will feed the masked item sequence (S_A^m) to the embedding layer for extracting the item representation. Since the encoder part of Transformer in our model cannot consider the order of the sequence. Therefore, the embedding layer in our model is the sum of two types of embedding which are token embeddings and positional embeddings. We will obtain the item representation called item sequence embedding. Next, we feed the item sequence embedding into the stack of Transformer for training and we will receive the final hidden layer output $\mathrm{H}_V = \{h_{v1}, h_{v2}, \dots, h_{vm}, \dots, h_{vt}\}$ where h_{vt} show the similarity on every pair of items in the sequence for each query item v_t. Then, we make the distribution of all items over the query item v_t by adding the feedforward layer with GELU activation as shown in Equation (28).

$$P(v_t) = softmax(GELU(h_{vt}W^h + b^h)E^V + b^f) \tag{28}$$

where W^h and b^h is the weight matrix and bias term of final hidden layer output H_V respectively. E^V is the embedding matrix for all items in the set V which have passed the token embeddings and positional embedding. And b^f is the bias term of GELU activation function.

The objective of the training step is to let the model be able to reconstruct the masked item as the original item as close as possible by predicting the masked item v_m

based on the final hidden layer output of this token h_{vm}. Then, measuring the loss function as the negative log-likelihood of the masked item as Equation (29).

$$L = \frac{1}{|M_A|}\sum_{v_m \in M_A} - logP(v_m = v_m^T | S_A^m) \quad (29)$$

where M_A is the set of randomly masked items in S_A^m, v_m^T is the true item for the masked item v_m, and P(.) is the probability that was defined in Equation (28).

After BERT model has finish training, we will receive the final hidden layer output H_V' which has the weight that can predict the next item for the target *userA*. Then, in the testing step, we feed the item sequence of target *userA* as an input and append a mask token, which is selected from set $V - S_A$ and put it at the end of the sequence. This mask token is the target item (v_{m_t}) and can assume as the new item that the target user has never seen before. After that, we feed the item sequence to the embedding layer and feed it into the finished training BERT model. The model will predict the target item based on the final hidden layer output h'_{vm_t}. Thus, we make the distribution of all items in the sequence over the target item as below

$$P(v_{m_t}) = softmax(GELU(h'_{vm_t}W^{h'} + b^{h'})E^V + b^f) \quad (30)$$

where $W^{h'}$ and $b^{h'}$ is the weight matrix and bias term of the final hidden layer output H_V', respectively, and E^V is the embedding matrix for all item in the set $V - S_A$.

Repeating the step that we described above to all new item that selected from the item set $V - S_A$. Finally, we will receive the probability of all new items that the target *userA* potential to interact with and we call it a target *userA* profile.

3.2 Collaborative Filtering

In the collaborative filtering side, our model aims to find who is the neighbor of the target user by measuring the similarity between the target user and other users in the system. To find who is the neighbor of the target user, we have adopted the BERT4Rec model to our model and feed the user sequence instead of the item sequence as an input, which is defined in the content-based filtering side. The user sequence is the sequence

of all users that used to rate the target item. The details of the step in the collaborative filtering side are as follows.

Initially, let $U = \{u_1, u_2, \ldots, u_{|U|}\}$ be the users set, the target item is *itemO* and $S_O = \{u_1, u_2, \ldots, u_t\}$ be the user sequence of target *itemO* where u_t is the user who used to rated *itemO* at time step t. In the same way as the content-based filtering side, we randomly mask 15% of all users in the sequence and replace it with "[mask]" token as illustrated in Figure 3.2 and define as $S_O^m = \{u_m, u_2, u_3, \ldots, u_t\}$ where u_m is the user that was replaced with the "[mask]" token. The only masked user will keep in M_O and in Figure 3.2 you can see that the masked user is u_1.

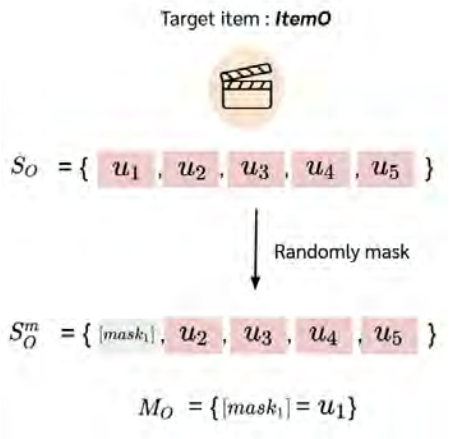


Figure 3.2 The input of collaborative filtering part

After random mask the user in the user sequence, we feed the masked user sequence (S_O^m) to the embedding layer which consists of token embeddings and positional embeddings and we will obtain the user representation called user sequence embedding. Then, we feed user sequence embedding to the stack of Transformer for training as same in the content-based filtering side and we obtain the final hidden layer output $H_U = \{h_{u1}, h_{u2}, \ldots, h_{um}, \ldots, h_{ut}\}$ where h_{ut} show the similarity on every pair of users in the sequence for each query user u_t. Next, we add the feedforward layer with GELU activation to produce the distribution of all users over the query user u_t as below

$$P(u_t) = softmax(GELU(h_{ut}W^h + b^h)E^U + b^f) \quad (31)$$

where W^h and b^h is the weight matrix and bias term of the final hidden layer output H_U respectively, and E^U is the embedding matrix for all users in the set U.

The objective of the training step in the collaborative filtering side is same as the content-based filtering side, we aim to make the model can reconstruct the masked user to be the original user as similar as possible. Then, we let the model predict the masked user u_m by using the final hidden layer output of u_m (h_{um}). And we define negative log-likelihood of the masked user as the loss function as in Equation (32).

$$L = \frac{1}{|M_O|}\sum_{u_m \in M_O} - logP(u_m = u_m^T | S_O^m) \tag{32}$$

where M_O is the set of randomly masked users in S_O^m, u_m^T is the true user for the masked user u_m, and P(.) is the probability that was defined in Equation (31).

After finished training, we have the final hidden layer output H'_U which has the weight that can predict who is the neighbor of the target user. In the testing step, we feed the user sequence of target *itemO* as an input and append the mask token at the end of the sequence which represents the target user (u_{m_t}). Then, feed the user sequence to the embedding layer after that feed it into the finished training BERT model. The model will predict the target user based on the final hidden layer output h'_{um_t}. Consequently, we make the distribution of all users in the sequence over the target user as below

$$P(u_{m_t}) = softmax(GELU(h'_{um_t}W^{h'} + b^{h'})E^U + b^f) \tag{33}$$

where $W^{h'}$ and $b^{h'}$ is the weight matrix and bias term of final hidden layer output H'_U, respectively, and E^U is the embedding matrix for all item in the set $U - S_O$.

Iterating the step that we mentioned above to all user that selected from user set $U - S_O$. Finally, we will receive the probability of all users which are similar to the target user and we call it a target *itemO* profile.

3.3 Prediction Stage

In the prediction stage we adopt the MF that applies in NCF to predict the rating in our model. NCF wants inputs that are representation of user and representation of item. Thus, we use target user profile and target item profile which are user representation and item representation respectively as NCF input. Finally, we receive a rating of the target user toward the target item.

CHAPTER IV
EXPERIMENTAL EVALUATION

In this chapter, the experimental results of the proposed method are compared with the BERT4Rec. BERT4Rec is the content-based filtering approach which only considers the historical data of the target user. In contrast, our proposed method consists of both content-based filtering and collaborative filtering approach which escalate to consider the preference of the target users' neighbors. Therefore, the organization of this chapter is as follows. First, the details of the dataset that was used in this experiment are explained. Next, the evaluation metrics in this experiment is introduced which is evaluated in terms of accuracy (NDCG). Finally, the experimental results of the proposed method and BERT4Rec are compared.

4.1 Datasets

To perform experiments, MovieLens-1M dataset is used. It consists of 6,040 users, 3,706 movies, and 1,000,209 rating records with rating range 1 to 5. The sample of MovieLens-1M dataset which is used in these experiments is shown in Table 4.1. It consists of 4 columns which are userId, movieId, rating, and timestamp. The first record means *userId1* rated *movieId1193* with rating score *5* at timestamp *978300760*.

Table 4.1 The sample from MovieLens-1M dataset

userId	movieId	rating	timestamp
1	1193	5	978300760
1	661	3	978302109
1	914	3	978301968
1	3408	4	978300275
1	2355	5	978824291

For data preprocessing, we have created item sequences for each user and user sequence for each item by grouping the records by users and items, respectively. Then, sorting the records according to the timestamp. After that, removing users that have less than 20 feedbacks and items that were rated less than 5 feedbacks from the dataset. The

sample of item sequence and user sequence are shown in Table 4.2 and Table 4.3, respectively.

Table 4.2 The sample of item sequence after preprocessing

userId	movieId
1	3186
1	1721
1	1022
1	1270
1	2340

Table 4.3 The sample of user sequence after preprocessing

movieId	userId
1	6035
1	6032
1	6022
1	6021
1	6016

4.2 Evaluation Metrics

In this work, we evaluate the performance of our model against BERT4Rec in terms of prediction accuracy. We adopt Normalized Discounted Cumulative Gain (NDCG) as our evaluation metrics in order to evaluate the accuracy of top-K recommendation ranking list for each user as is demonstrated by

$$NDCG_K = \frac{DCG_K}{IDCG_K} \tag{34}$$

where DCG_K denotes discounted cumulative gain which can be computed by Equation (35), and $IDCG_K$ denotes ideal discounted cumulative gain which is the possible highest value of DCG_K among the ranking list of items.

$$DCG_K = \sum_{i=1}^{K} \frac{2^{rank_i - 1}}{\log_2(i+1)} \quad (35)$$

where $rank_i$ = {5,4,3,2,1} is the actual rating score of an item at the top-K rank position i.

Table 4.4 Actual rating and predicted rating of target user u

ItemId	Actual rating	Predicted rating
1	2	3
2	4	3
3	4	2
4	3	2
5	5	2
6	5	4

To comprehend more about NDCG, for the target user u, let actual rating and predict the rating of the target user u is shown in Table 4.4. To calculate NDCG with K=3, firstly Table 4.4 is sorted by actual rating and predict rating as illustrated in Table 4.5 and assume that this table is the ranking list of items.

Table 4.5 Sorted actual rating and predicted rating of target user u

Sorted ItemId	Actual rating	Sorted ItemId	Predicted rating
5	5	6	4
6	5	1	3
2	4	2	3
3	4	3	2
4	3	4	2
1	2	5	2

Then, the ranking list of items from actual ratings to generate the position index and prediction rating. Instead of using a general position index of rating, we change position index by if the item has the same rating, it will have the same position index as

shown in Figure 4.1. In Table 4.6 we show generation prediction rating. For items which are not in the ranking list of items from actual ratings, prediction ratings are replaced by zero, in contrast for items in the ranking list of items from actual ratings, we will check items' position index in the ranking list of items from predicted ratings. Next, replacing prediction ratings with ratings in actual ratings at the same position index.

Actual rating	5	5	4
Index	1	1	3

Figure 4.1 The position index top-3 ranking list

Table 4.6 Actual rank rating and predicted rank rating of target user u

Actual rank ItemId	Actual rank rating	Predicted rank ItemId	Predicted rating	Predicted rank rating
5	5	6	4	0
6	5	1	4	5
2	4	2	3	4

After that computed DCG_3 and $IDCG_3$ by considering the data with pink highlight and yellow highlight in Table 4.6, respectively. The rating 0, 5, and 4 in pink highlight are the top-3 predicted rank rating that we generate from the actual rating. The rating 5, 5, and 4 in yellow highlight are the top-3 actual or ideal rank rating score that is relevant to the item 5, 6, and 2. Finally, using DCG_3 and $IDCG_3$ to compute $NDCG_3$. The solution of how to compute DCG_3, $IDCG_3$ and $NDCG_3$ is shown in Figure 4.2.

$$DCG_3 = \frac{2^0 - 1}{\log_2(1+1)} + \frac{2^5 - 1}{\log_2(1+1)} + \frac{2^4 - 1}{\log_2(3+1)} = 38.5$$

$$IDCG_3 = \frac{2^5 - 1}{\log_2(1+1)} + \frac{2^5 - 1}{\log_2(1+1)} + \frac{2^4 - 1}{\log_2(3+1)} = 69.5$$

$$NDCG_3 = \frac{DCG_3}{IDCG_3} = \frac{38.5}{69.5} = 0.554$$

Figure 4.2 The solution of computing DCG_3, $IDCG_3$ and $NDCG_3$ of target user u

Since All users in MovieLens-1M dataset have rated items more than ten times, we select $NDCG_5$, $NDCG_7$, $NDCG_{10}$ to evaluate our model.

4.3 Experimental Results

To evaluate the experiment of our proposed method, we compare it with BERT4Rec in terms of accuracy which is the NDCG. For environment setting, we use GPU (RAM: 25.5 GB, Disk: 69.4 GB). In this experiment we split the dataset into 80:20 which 80% is training data and 20% is test data. We use this data for both BERT4Rec and our proposed method (HybridBERT4Rec). Training time around 17 seconds per epoch and test time around 5 seconds per epoch. In addition, we vary the top-K recommendation ranking list to 5, 7, and 10. The experimental results are presented in the following.

Table 4.7 Performance comparison of our proposed method and BERT4Rec

	MovieLens-1M		
	$NDCG_5$	$NDCG_7$	$NDCG_{10}$
BERT4Rec	0.9461	0.8249	0.6929
HybridBERT4Rec	**0.9942**	**0.8903**	**0.8134**

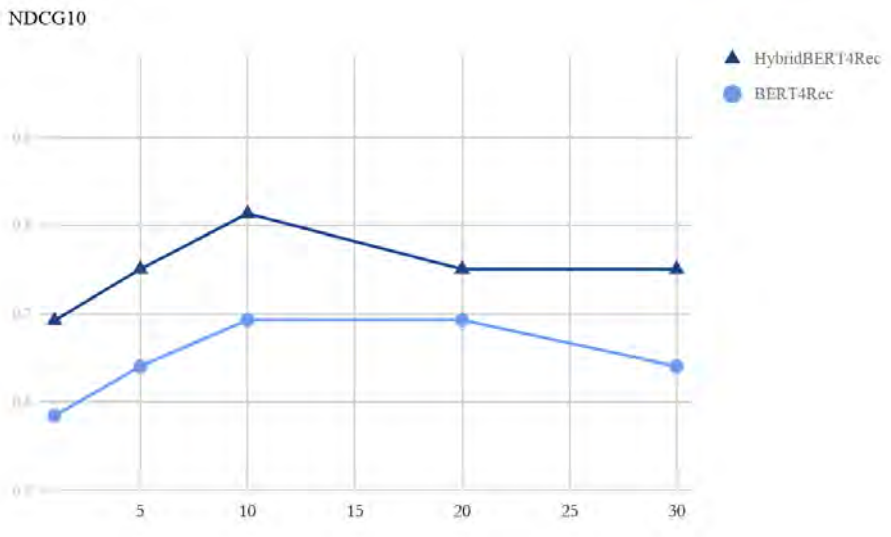


Figure 4.3 $NDCG_{10}$ of our proposed method and BERT4Rec at 1, 5, 10, 20, 30 epochs

From result in Table 4.7, it can be shown that our proposed method has results of $NDCG_5$, $NDCG_7$, $NDCG_{10}$ higher than BERT4Rec. Since BERT4Rec has only content-based filtering but our proposed method appends collaborative filtering by using neighbor of the target user for cooperation in recommendations.

CHAPTER V
CONCLUSION

In this chapter, the conclusion of the proposed method is presented and the future plan of this work is explained for improving the proposed method to have more efficiency in the recommender system.

5.1 Conclusion

In this work, we proposed a Hybrid (Content-based filtering and Collaborative filtering) recommender system based on BERT to solve the serendipitous problem of the BERT4Rec model and allow BERT to consider the interaction of other users in the system. In the content-based filtering side, we feed the item sequence of the target user as an input of BERT model and we receive the target user profile. In the collaborative filtering side, we feed the user sequence which is the sequence of users who used to rate the target item, as an input of BERT model and we obtain the target item profile. After receiving the target user profile and target item profile, we use it as an input of NCF model for predicting the rating score. From the experiment results that were presented in chapter IV, it can be concluded that our proposed method outperforms BERT4Rec model in terms of accuracy.

5.2 Future Plan

From the experiment results that our proposed method outperforms BERT4Rec model, we will try to experiment on other datasets such as MovieLens-20M or Amazon Beauty dataset and so on. And since we use only one evaluation metric in this experiment, we aim to use some other evaluation metrics to evaluate our proposed method in the future.

REFERENCES

[1] Ashishkumar , P., Kiran, A. 2018. Serendipity in Recommender Systems. **International Journal of Engineering and Technology (IJET).** 10(1) : pp. 202-206.

[2] Cui, Q., Wu, S., Liu, Q., Zhong, W., Wang, L. 2018. MV-RNN: A Multi-View Recurrent Neural Network for Sequential Recommendation. **IEEE Transactions on Knowledge and Data Engineering.** 32(2) : pp. 317-331.

[3] Devlin, J., Chang, M., Lee, K., and Toutanova, K. 2018. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.** arXiv preprint, arXiv : 1810.04805.

[4] Harper, M., Konstan, A. J. **MovieLens 1M Dataset** [Online]. Available from : https://grouplens.org/datasets/movielens/1m/ [2020, January 14].

[5] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. 2017. Neural Collaborative Filtering. **Proceedings of the 26th International Conference on World Wide Web**, pp. 173-182. Perth, Australia : The International World Wide Web Conference Committee (IW3C2).

[6] Koren, Y., Bell, R., Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. **IEEE Computer**. 42(8) : pp. 30-37.

[7] Oord, A., Dieleman, S., Schrauwen, B. 2013. Deep Content-Based Music Recommendation. **Proceedings of the Neural Information Processing Systems Conference (NIPS)**, pp. 2643-2651. Curran Associates, Inc.

[8] Pennington, J., Socher, R., Manning, D. C. 2014. GloVe: Global Vectors for Word Representation. **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1532-1543. Doha, Qatar : Association for Computational Linguistics.

[9] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. **Proceedings of the 28th ACM International Conference on Information and Knowledge Management**, pp. 1441-1450. New York, United States : Association for Computing Machinery (ACM).

[10] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. 2015. Going deeper with convolutions. **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 1–9.

[11] Tang, J., Wang, K. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. **Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining**, pp. 565-573. Marina Del Rey, CA, USA : Association for Computing Machinery (ACM).

[12] Taylor, L. W., 1953. Cloze procedure: A new tool for measuring readability. **Journalism Bulletin**. 30(4) : pp. 415-433.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, N. A., et al. 2017. Attention Is All You Need. **Proceedings of the Neural Information Processing Systems Conference (NIPS)**, pp. 5998-6008. Curran Associates, Inc.

[14] Wu, Y., Schuster, M., Chen, Z., Le, V. Q., Norouzi, M., Macherey, W., et al. 2016. **Google's neural machine translation system: Bridging the gap between human and machine translation.** arXiv preprint, arXiv : 1609.08144.

[15] Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. **Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval**, pp. 729-732. Pisa, Tuscany, Italy : Association for Computing Machinery (ACM).

APPENDICES

APPENDIX A

The Project Proposal of Course 2301399 Project Proposal

Academic Year 2019

Project Tittle (Thai)	ระบบแนะนำผู้ใช้แบบผสม (การกรองเนื้อหาและการกรองแบบร่วมมือกัน) โดยมีพื้นฐานอยู่บน BERT
Project Tittle (English)	Hybrid (Content-based filtering and Collaborative filtering) recommender system based on BERT
Project Advisor	1. Assoc. Prof. Saranya Maneeroj, Ph.D. 2. Asst. Prof. Kitiporn Plaimas, Ph.D.
By	1. Chanapa Channarong ID 5933616323 2. Chawisa Paosirikul ID 5933618623 Computer Science Program, Department of Mathematics and Computer Science Faculty of Science, Chulalongkorn University

---

Background and Rationale

A recommender system plays an essential role in our everyday life in which we have seen on many websites and applications like YouTube, Facebook, Netflix, and Amazon. Furthermore, various technologies were developed. It makes people be able to access much information that leads to information overload. Therefore, the recommendation system is an essential part of supporting user decisions.

Collaborative filtering is one of the primary methods commonly used in the recommender system. This method measures the similarity between a target user and its neighboring users, and calculates the rating score to create a list of recommended items for the target user. However, there is the cold start user problem when a new user enrolls in the system since this user will have incomplete information. In other words, he/she has never give ratings to items or give rating only through small number of items. This problem causes a poor-quality recommendation. This type of the recommender system also causes the sparse data. Mostly, the number of items is greater than the users so it is hard to calculate similarities between users. For this reason, this method requires a large amount of data to calculate the recommendation.

Another primary method is Content-based filtering, which measures the similarities between an item that is going to be recommended for a target user, and items that the target user was familiar with in the past by considering the content that describes the items. The advantage of this method is that it does not require many data for processing or there is no cold start. But the limitation is that the set of items that were recommended for a target user are quite specific because it cannot recommend new items that the target user has never seen before.

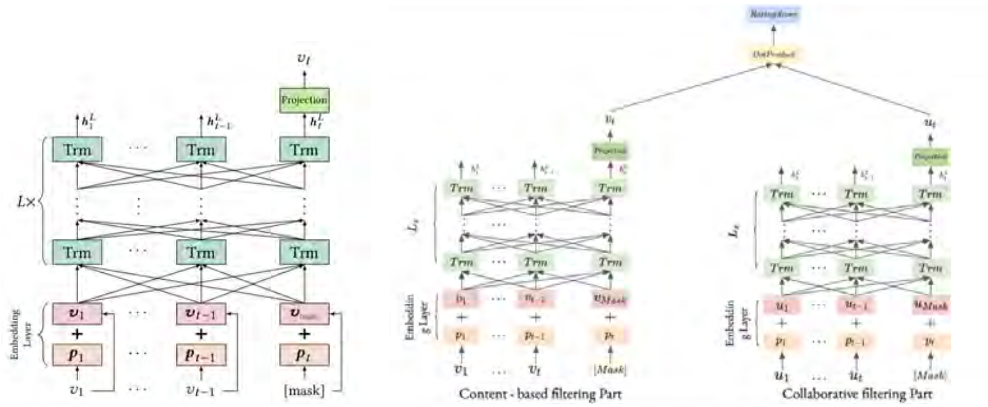
However, the two methods above have both advantages and disadvantages. Therefore, they were combined to improve the recommender system to be more efficient but still trying to preserve the advantages and eliminate some disadvantages — this is called hybrid recommender system approach.

BERT stands for Bidirectional Encoder Representations from Transformers. It is a model that brings the encoder parts of the Transformer model [2] to generate a language model. The regular Transformer model has limitations to train in bidirectional and to use the fully self-attention method. Therefore, BERT applied bi-direction training to the regular Transformer model. However, language modeling are trained left-to-right by predicting the next word of each position in the input sequence. Jointly context on both left and right in bi-direction training would cause information leakage which allows each word to see the target word hence the model would not learn anything useful. BERT adopts masked language modeling, which randomly masks words and assigns the model to predict that word based on their surrounding context.

The earlier work [3] adopts the deep bidirectional self-attention model BERT to the sequential recommender system for predicting the next item that users are likely to interact with. By giving the historical interaction and applying masked language modeling [4] randomly on the items, then predicted the masked items based on the surrounding items. The previous model extracts the user profile by considering the similarity on every pair of items in his/her historical item sequences. When the new item has entered, this model will predict which item should be the next (target item). Moreover, their model only use historical data of a target user without considering interaction information of other users toward a target item which is the content-based filtering approach. It will be better if collaborative filtering can be applied in BERT for making their model to have higher accuracy that is the Hybrid recommendation approach. By looking at the information of other users that interact with a target item.

Instead of an attention on item sequences of the target user, we are interested in user sequences. Users in this sequence are the users who used to rate or interact with the target item. In order to extract which users affect to the target item and would be neighbors of the target user, we will apply attention mechanism in BERT by feeding another input which is the user sequences of the target item.

In this work, we propose a Hybrid (Content-based filtering and Collaborative filtering) recommender system that applying BERT model. In addition to content-based in the previous model, we intend to escalate user-based collaborative filtering by considering the other users in the system that have interactions with the target item. From the previous model in Figure (a), it shows that it has employed BERT on only item sequences rated by the target user that is the content-based filtering approach, while our model in Figure (b) incorporates a collaborative filtering approach by feeding another input sequence including all users who used to rate the target item into the model. After taking input as item sequence (content-based filtering part) and user sequences (collaborative filtering part), we will obtain target user profile and target item profile respectively. For target user profile, it consists of the similarity on every pair of items on historical sequence of the target user. Consequently, it provides the information whether the next item is the target item or not. Meanwhile, a target item profile consists of the similarity on every pair of users which show who are the neighbors of the target user. In the prediction state, we apply NCF [4] to predict the rating score between target user profile and target item profile.



(a) BERT4Rec model architecture (b) hybridBERT4Rec model architecture

Objectives

1. To propose a new method that applied BERT on both Content-based filtering approach and Collaborative filtering approach.
2. To compare performance in terms of accuracy between proposed model and traditional model (BERT4Rec [8]).

Scope

1. Use the MovieLens- dataset that consist of 610 users, 9,742 movies and 9,724 ratings. The lowest rating score is 0.5 and the highest rating score is 5.0
2. Compare the efficiency of the proposed approach with the research of Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang about BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer.

Project Activities

1. Outline of study
   1.1 Study the architecture of the current recommendation system.
   1.2 Study the research and academic articles in recommendation systems.
   1.3 State the pros and cons of the previous methods.
   1.4 Analyze and design methods to solve the problem.
   1.5 Develop a correctness test of the proposed method.
   1.6 Perform an experiment to measure the performance of the proposed system.
   1.7 Analyze and discuss the experimental results.
   1.8 Summarize and write a report.
2. Timeline of study

   From the outline of study above, we can write the Gantt chart as below.

Procedures	2019						2020			
	07	08	09	10	11	12	01	02	03	04
1. Study the architecture of the current recommendation system.										
2. Study the research and academic articles in recommendation systems.										
3. State the pros and cons of the previous methods.										
4. Analyze and design methods to solve the problem.										
5. Develop a correctness test of the proposed method.										
6. Perform an experiment to measure the performance of the proposed system.										
7. Analyze and discuss the experimental results.										
8. Summarize and write a report.										

Benefits

1. For Researcher

   1.1 Learn the operation of recommender systems.

   1.2 Learn the theory and practice in creating a new approach of recommender systems.

   1.3 Apply knowledge to create a recommender system.

   1.4 Practice the skill of work planning

   1.5 Practice in solving problems that occur during the operation.

2. For Business and Society

   2.1 Develop knowledge that will be applied to a business in the future.

   2.2 Continue to develop a new knowledge which is beneficial to the research industry.

Equipment

1. Hardware
   1.1 Computer with macOS Catalina version 10.15 with Processor 2 GHz Intel Core i5 Memory 8 GB and Storage 256 GB.
   1.2 Computer with macOS Catalina version 10.15 with Processor 1.4 GHz Intel Core i5 Memory 8 GB and Storage 256 GB.
2. Software
   2.1 PyCharm Professional version 2019.2.3
   2.2 Visual Studio Code version: 1.39.0

Budget

1.	Apple Magic Mouse 2	2	4,580 baht
2.	LaCie HDD External Mobile Drive 2TB STHG2000400	1	2,590 baht
3.	WDD HDD 4TB My Passport Ultra Type-C USB 3.0	1	3,290 baht
		Total	10,460 baht

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805.*

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *arXiv preprint arXiv:1706.03762.*

[3] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. *arXiv preprint arXiv:1904.06690.*

[4] Wilson L Taylor. 1953. Cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

BIOGRAPHY

Miss Chanapa Channarong
Department of Mathematics and Computer Science
Faculty of Science, Chulalongkorn University
Email: chanapa_parn@hotmail.com
Interested Field: Recommender System, Machine Learning

Miss Chawisa Paosirikul
Department of Mathematics and Computer Science
Faculty of Science, Chulalongkorn University
Email: chawisa_gieza@hotmail.com
Interested Field: Recommender System, Machine Learning