Movie Recommendation System: A Hybrid Approach

Bandaru Bhavyanath
B180399EC

Chevuru Sai Jaswanth
B180668EC

Eruguri Jeshwanth
B180556EC

Jarapala Mohith Pamar
B181027EC

Under the co-guidance of
Dr. Vinay Joseph
Assistant Professor,
ECE Department, NITC.

Under the co-guidance of
Dr. Nujoom Sageer Karat
Faculty Member,
ECE Department, NITC.

***Abstract*—Natural language processing is crucial in today's progressing world because it helps overcome linguistic ambiguity like speech recognition and text analytics. Natural language processing has made great progress recently using the BERT framework. Recommendation System is another significant area that is very popular and useful faster-automated decisions. Surveys show we spend more than 100 days of our lives choosing what to watch. A Movie Recommendation System can suggest a set of movies to users based on their preferences or the popularities of the movies. To improve the quality of recommendations, we have developed a hybrid approach by combining collaborative filtering using Matrix Factorization [1] and content-based filtering using BERT4Rec [2] which is an extension of the BERT framework for recommendation systems. Several studies indicate that the hybrid approach can provide more accurate recommendations. We have utilized the MovieLens dataset, which is a representative and popular real-world benchmark dataset in recommendation systems. We have evaluated the performance of our novel movie recommendation system.**

***Index Terms*—Recommendation System, Collaborative filtering, Content based filtering, BERT4Rec, Mean Average Precision.**

I. Introduction

Natural Language Processing (NLP) is an area of artificial intelligence that deals with natural language interaction between computers and humans. NLP is progressing quickly, thanks to a growing interest in human-machine communication, as well as the availability of massive data, powerful computation, and improved algorithms. The development of BERT changed the NLP paradigm significantly. BERT4Rec [2] is the first to utilize the BERT framework for sequential recommendation tasks, achieving state-of-the-art performances. Hybrid recommender systems aim to model users' profiles for personalized recommendations. They also help users find the movies of their choice based on the movie experience of other users efficiently and effectively.

II. Problem Statement

Our problem is to design and assess a novel movie recommendation system that utilizes a hybrid approach and applies the BERT4Rec framework, which is typically used in NLP settings, to recommendation systems. The hybrid approach enables us to investigate the performance of BERT4Rec combined with collaborative filtering, and avoid disadvantages of pure approaches such as the lack of information about the domain dependencies in collaborative filtering and people's preferences in content-based systems.

III. Our approach

- Preprocess the dataset to extract all the relevant information for the design and execution of recommender models.
- Achieve collaborative filtering using Matrix Factorization [1] model. The immediate output is a list of movies recommended to a user based on historical interactions of the similar users.
- Achieve content based filtering using BERT4Rec [2] model. The output is a list of recommendations to the user.
- Design a hybrid model by pipelining both the above recommendation systems.
- Evaluate all our models to fine-tune the performance.

A. Tools

1) Packages:: We plan to use NumPy, Pandas, PyTorch to develop the hybrid model using Google Colab.

2) Dataset:: MovieLens is a popular benchmark dataset for evaluating recommendation algorithms. This dataset contains user's ratings of movies, as well as timestamps. We have used ML-1m variant.

3) Metrics:: Evaluation of our model using python ml-metrics library to achieve Mean Average Precision.

IV. Collaborative Filtering

Collaborative filtering (CF) recommender systems create predictions based on user-item relationships, with no further knowledge about the users or objects required.

The user-item rating matrix is used in latent factor models for movie recommendations to profile both users and items using a number of latent variables. A feature in the context of movies can be a genre, target age range, or even something utterly unintelligible. Matrix factorization is an excellent latent factor approach for recommender systems.

A. Matrix Factorization

Matrix Factorization (MF) [1] is a method for calculating a system's latent factor model based on user-item interactions represented as a matrix. Even on relatively sparse matrices, Matrix Factorization has been shown to provide very strong predictions.

By factoring the rating matrix into a product of two latent component matrices, the matrix factorization approach decreases the dimensions of the rating matrix. Refer figure 2 for depiction of latent component matrices.

B. Matrix Factorization with bias

To improve the predictions, use a bias for movie i, called b_i, bias for user u, called b_u to model the rating r_{ui}.

$$r_{ni} = b_i + b_n + u_n m_i^T \tag{1}$$

In addition to learning the bias independently, This can be solved using a stochastic gradient descent approach.

Algorithm: The algorithm using Stochastic Gradient Descent was implemented as follows:

1) Initialize matrices u and m of sizes users $\times K$ and Movies $\times K$ with random values from a uniform distribution over [0,0.05] where K is the number of features that will be extracted.
2) Iterate over all the observed ratings in training dataset.
   a) Calculate a predicted rating.
   b) Calculate error for the predicted rating to that of actual rating using Mean Square Error.
   c) Update u and m according to error with the help of Adam Optimizer.
3) Calculate the model learned ratings by taking dot product of user and item embeddings, and arrange them in descending order by value.

V. Content Based Filtering

Content-based filtering leverages item similarity to suggest items that are related to what a user enjoys. Content-based filtering generates suggestions by matching keywords and attributes assigned with user profiles.

A. Transformer

Transformer is a novel architecture that exhibits feed-forward neural network behavior and adopts the self-attention mechanism. Sequential computation prevents parallelization. RNNs are feedback neural networks which require temporal dependency on weights. RNNs, LSTMs and GRUs still need more-prominent attention mechanism to address the long-range dependencies. Transformers paved way to modern pre-trained models like BERT. Refer Figure 3 for the transformer architecture.

B. BERT

Bi-directional Encoder Representations from Transformers (BERT) is stack of certain number of transformer encoder layers, mainly pre-trained on masked language modelling (MLM) and next sentence prediction (NSP). Fine-tuning BERT can address tasks like Neural Machine Translation, Question Answering, Sentiment Analysis, Text Summarization, etc.

C. BERT4Rec

BERT4Rec is a sequential recommendation model that uses deep bidirectional self-attention to model user behavior sequences. Previous models using sequential NN's are sub-optimal due to flaws such as:

(a) Unidirectional architectures limit the power of hidden representation in user's behavior sequences.

(b) They often assume a rigidly ordered sequence, which is not always feasible.

Refer Figure 4 for BERT4Rec architecture.

Training: BERT4Rec model is trained using Masked Language modelling (MLM). We send the user watch history, masking with a certain probability. The model predicts the masked movies based solely on its bidirectional context, and calculates the loss by comparing with the hidden representations.

Loss function: We define the loss for each masked input as the negative log-likelihood of the masked targets.

$$\mathcal{L} = \frac{1}{|\mathcal{S}_u^m|} \sum_{v_m \in \mathcal{S}_u^m} -\log P\left(v_m = v_m^* \mid \mathcal{S}_u'\right) \tag{2}$$

Testing: Append the mask token to the end of user's behavior sequence, and then predict the next item based on the final hidden representation of this token.

VI. Evaluation

A. Matrix Factorization

The most recent Movie-Lens dataset was used, which included 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 Movie-Lens users. The data for the ratings was split as 70:10:20 between training, validation and testing. The training and validation losses vs epochs graph is plotted, refer Figure 5.

Mean Squared Error: The loss function (L) adopted in the model is the Mean Square Error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

We have used MSE to calculate the cost of our model predictions, based on test data.

Cosine Similarity: The most recent highest rated movie of a particular user is taken from the training dataset, and the cosine similarity score between that movie and his corresponding predictions is calculated, just as sanity check for the predictions given by our MF model.

B. BERT4Rec

We have used ML-1m dataset to train the model and the corresponding training and validation losses have been plotted. Refer Figure 6.

Precision: It is the ratio of recommended items that are relevant to the total number of recommended items.

Recall: It is the ratio of recommended items that are relevant to the total number of relevant items.

Average Precision: AP is a measure of the number of relevant products recommended in the first "K" results. AP is calculated using precision and recall at every position in the ranked sequence of recommendations.

$$AP = \sum_{1}^{x} (precision\ at\ i) \times (change\ in\ recall\ at\ i) \quad (4)$$

Mean Average Precision: MAP is the average of Average Precisions for all users.

$$MAP = \frac{\sum_{u=1}^{N} AP_u}{M} \quad (5)$$

MAP@K shows the MAP for top K recommendations.

C. Hybrid Model

Pipe-lining both the collaborative filtering and content based filtering models in a cascade fashion to design a novel hybrid recommendation system.

Cascade system: The recommendations of one technique are refined by another recommendation technique. The first recommendation technique outputs a coarse list of recommendations which is in turn refined by the next recommendation technique. Refer Figure 7 for the cascade system architecture.

VII. Results

A. Matrix Factorization

Initially we have used 15% of ML-1m dataset for testing and observed that the Mean Square Error is 0.756. Refer figure 8 for the MF model input and recommendations.

Using 20% of ML-1m dataset for testing, we have observed that the Mean Square Error = 0.760. Refer figure 9.

A significant increase in the no. of rating entries didn't majorly affect the model efficiency.

As a result, we were able to get recommendations of the most relevant movies to a user based on similar movie interests of other users.

B. BERT4Rec

We have taken a list of movies from the test dataset as the input and received a list of recommendations by the model.

The evaluation metric employed is Mean Average Precision, by setting one last movie as ground truth in the test data. This is known as Leave one out evaluation. Refer Figure 10 for the input, output and MAP score of the BERT4Rec model.

C. Hybrid Model

We have considered a random User ID as input to the pipeline model and the output is a list of predictions to that user. We have extended the same idea of Mean Average Precision evaluation implemented for BERT4Rec model, to the hybrid model and observed a significant increase in the MAP score. We have observed approximately 18% increase in the MAP score which indicates that the hybrid recommender system is enhancing the predictions for a user. Refer Figure 11 for the input, output and MAP score of the hybrid model.

VIII. Conclusion

Designed and assessed a novel hybrid movie recommendation system that utilizes a cascade approach and applies the BERT4Rec framework along with Matrix Factorization. We evaluated the collaborative and Content based models individually and designed the hybrid model. The hybrid system significantly improved the results. The hybrid approach enabled us to investigate the performance of BERT4Rec combined with collaborative filtering, and avoid disadvantages of pure approaches such as the lack of information about the domain dependencies in collaborative filtering and people's preferences in content-based systems.

Future prospects include more options for embedding keywords in BERT4Rec model, that allows to add items such as genres of a movie, plot, cast, etc. to further enhance the model's performance.

.

References

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[2] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1441–1450. [Online]. Available: https://doi.org/10.1145/3357384.3357895

APPENDIX

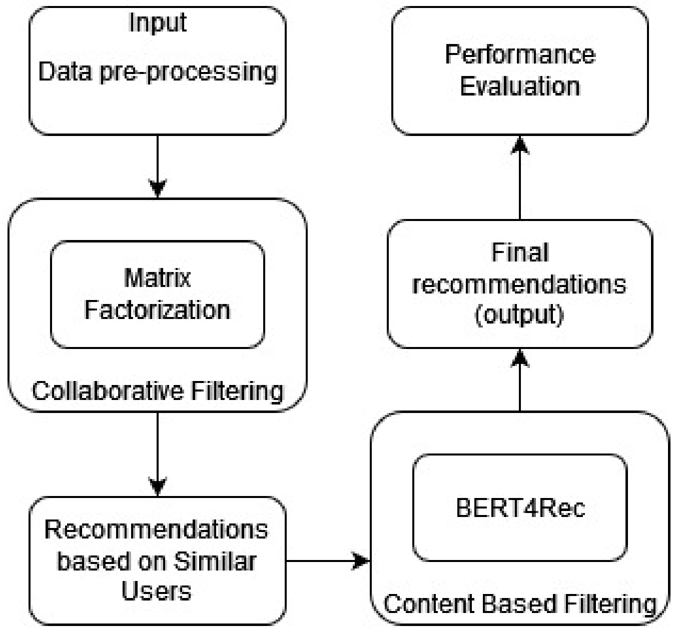


Fig. 1. Abstract Architecture of the model

$$\begin{bmatrix} r_{11} & . & . & . & r_{1i} \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ r_{n1} & . & . & . & r_{ni} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ . \\ . \\ . \\ u_n \end{bmatrix} \times \begin{bmatrix} m_1 & m_2 & . & . & . & m_i \end{bmatrix}$$

$$r = um^T$$

Fig. 2. Latent component matrices

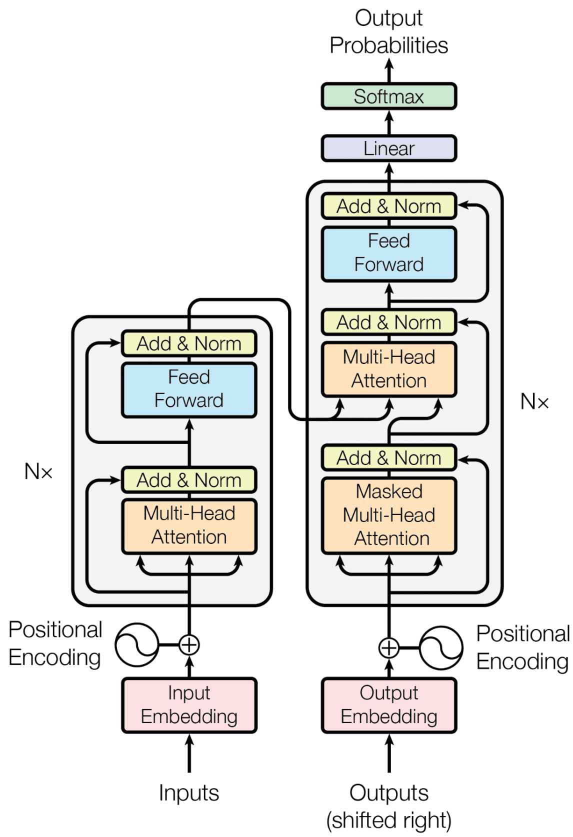
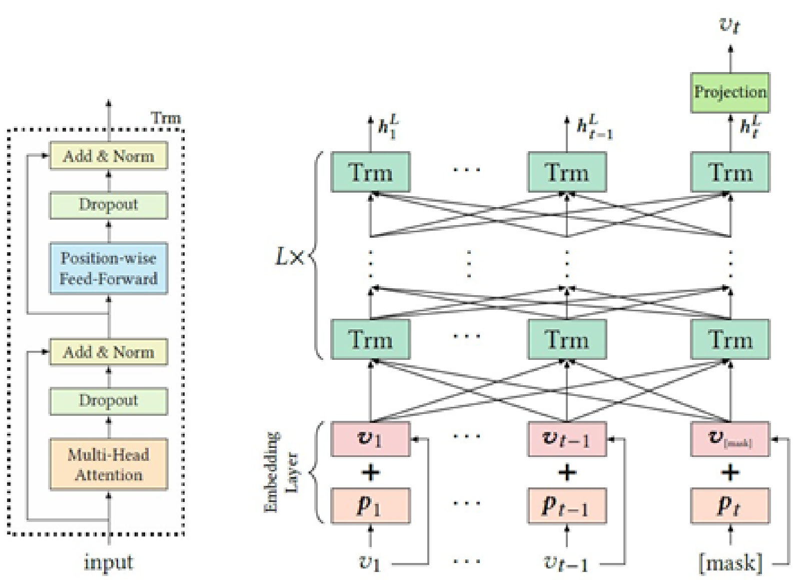


Fig. 3. Transformer Architecture (Source)



Transformer Layer. BERT4Rec model architecture.

Fig. 4. BERT4Rec Architecture [2]

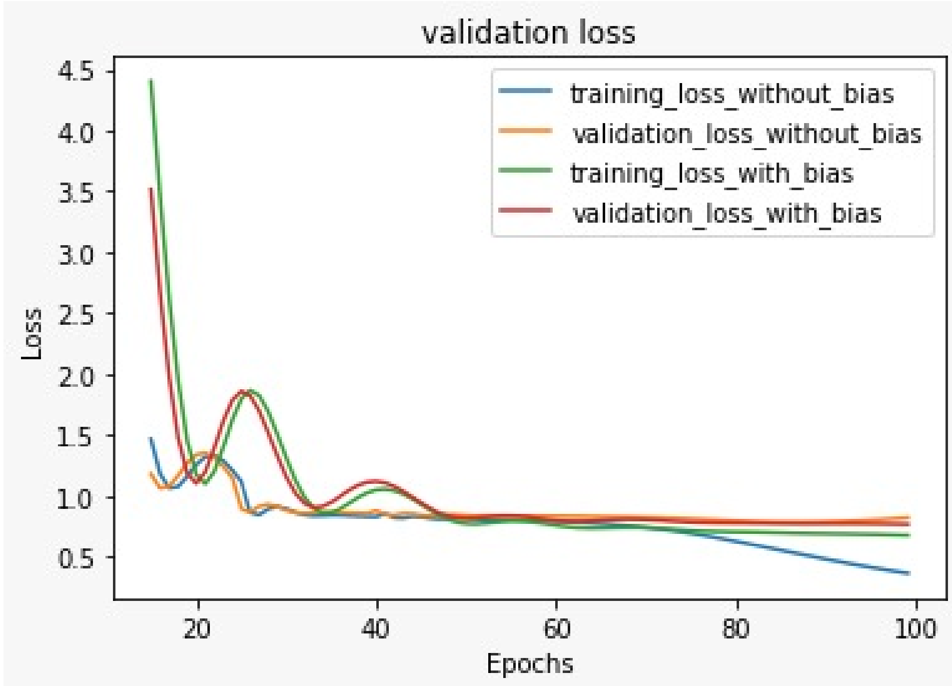


Fig. 5. Training and Validation Losses vs Epochs

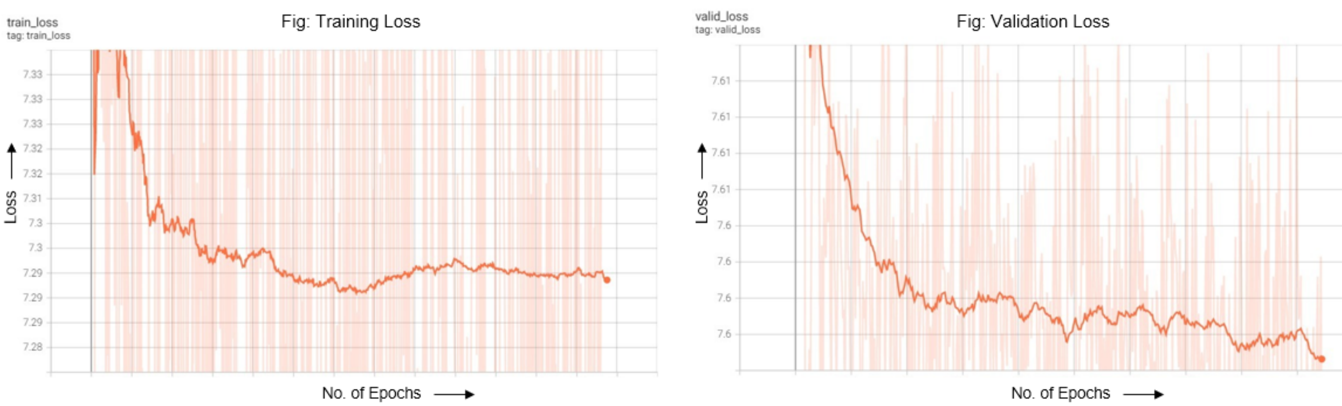


Fig. 6. BERT4Rec Model loss plots

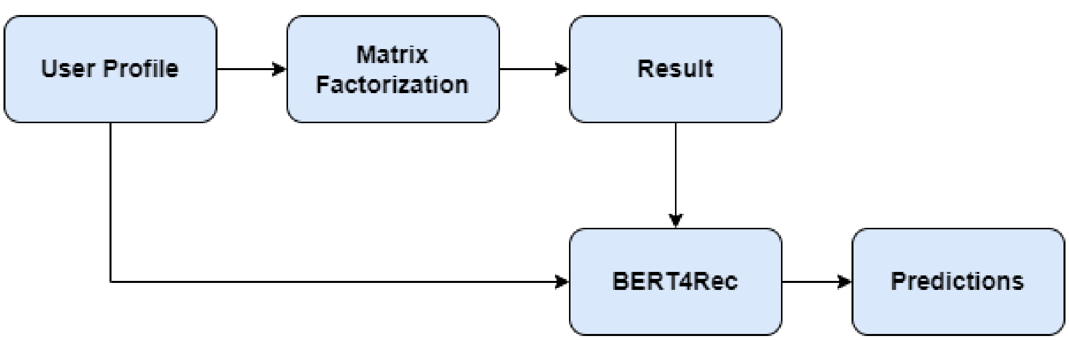


Fig. 7. Cascade approach

Using 15% of ML-1m dataset for testing

	userId	movieId	rating	timestamp	title
203635	270	258	4	968625006	Crocodile Dundee (1986)
382860	270	21	3	972669122	Leaving Las Vegas (1995)
466359	270	757	3	972669253	Little Shop of Horrors (1986)
878886	270	1030	3	968622946	Addams Family Values (1993)
741661	270	1860	4	968624035	Blame It on Rio (1984)
...	...	...	...	...	...
433529	270	380	5	968624602	Charlotte's Web (1973)
259151	270	783	4	972669198	Liar Liar (1997)
515253	270	623	3	968625788	Excalibur (1981)
552852	270	2294	4	968623942	Billy Madison (1995)
746837	270	1404	4	968623942	Big Trouble in Little China (1986)

[379 rows x 5 columns]
Most recent high rated movie by the user : Mr. Smith Goes to Washington (1939) 5

Fig: User watch history

```
Godfather, The (1972) --> 0.9203383
Schindler's List (1993) --> 0.90112156
Rear Window (1954) --> 0.9537657
Raiders of the Lost Ark (1981) --> 0.9158468
To Kill a Mockingbird (1962) --> 0.9560515
Sanjuro (1962) --> 0.9369534
Usual Suspects, The (1995) --> 0.889417
Lawrence of Arabia (1962) --> 0.94500756
Sixth Sense, The (1999) --> 0.8777086
Casablanca (1942) --> 0.960777
```

Fig: Recommendations given by our MF model

```
[50] print(mean_squared_error(model,test))

     0.7568132776446861
```

Fig: Mean Squared Error value

Fig. 8. Input, output, and MSE for 15% testing data

Using 20% of ML-1m dataset for testing

	userId	movieId	rating	timestamp	title
203635	270	258	4	968625006	Crocodile Dundee (1986)
382860	270	21	3	972669122	Leaving Las Vegas (1995)
466359	270	757	3	972669253	Little Shop of Horrors (1986)
878886	270	1030	3	968622946	Addams Family Values (1993)
741661	270	1860	4	968624035	Blame It on Rio (1984)
...	...	...	...	...	...
433529	270	380	5	968624602	Charlotte's Web (1973)
259151	270	783	4	972669198	Liar Liar (1997)
515253	270	623	3	968625788	Excalibur (1981)
552852	270	2294	4	968623942	Billy Madison (1995)
746837	270	1404	4	968623942	Big Trouble in Little China (1986)

[356 rows x 5 columns]
Most recent high rated movie by the user : Mr. Smith Goes to Washington (1939) 5

Fig: User watch history

```
Star Wars: Episode IV - A New Hope (1977) --> 0.94514924
Raiders of the Lost Ark (1981) --> 0.9559082
Matrix, The (1999) --> 0.8439266
Star Wars: Episode V - The Empire Strikes Back (1980) --> 0.94196665
Braveheart (1995) --> 0.8097833
Terminator, The (1984) --> 0.93777704
Star Wars: Episode VI - Return of the Jedi (1983) --> 0.93263555
Die Hard (1988) --> 0.9440742
Great Escape, The (1963) --> 0.9228151
Saving Private Ryan (1998) --> 0.87496656
```

Fig: Recommendations given by our MF model

```
[51] print(mean_squared_error(model,test))

     0.7604294228337461
```

Fig: Mean Squared Error value

Fig. 9. Input, output, and MSE for 20% testing data

Fig: BERT4Rec Input

```
['Mummy, The (1959)',
 "Mummy's Curse, The (1944)",
 'Message in a Bottle (1999)',
 'McCabe & Mrs. Miller (1971)',
 'Mary Poppins (1964)',
 'Man with the Golden Gun, The (1974)',
 'Man Who Would Be King, The (1975)',
 'Maltese Falcon, The (1941)',
 'Mad Max (1979)',
 'Lolita (1997)',
 'Lethal Weapon (1987)',
 'League of Their Own, A (1992)',
 'Goldfinger (1964)',
 'Ghostbusters (1984)',
 'Getaway, The (1994)',
 'Gaslight (1944)',
 'From Dusk Till Dawn (1996)',
 'Friday the 13th Part VII: The New Blood (1988)',
 'Friday the 13th Part V: A New Beginning (1985)',
 'French Connection, The (1971)',
 'Frankenstein (1931)',
 'Foxfire (1996)',
 'Foreign Correspondent (1940)',
 'For a Few Dollars More (1965)',
 'Force 10 from Navarone (1978)',
 'Fight Club (1999)',
 'Fighting Seabees, The (1944)',
 'Falcon and the Snowman, The (1984)',
 'Face/Off (1997)',
 'Exorcist II: The Heretic (1977)']
```

Fig: BERT4Rec Output

```
["One Flew Over the Cuckoo's Nest (1975)",
 'Aliens (1986)',
 'Alien (1979)',
 'Terminator, The (1984)',
 'Gladiator (2000)',
 'Stand by Me (1986)',
 'Wizard of Oz, The (1939)',
 'Shakespeare in Love (1998)',
 'Star Wars: Episode I - The Phantom Menace (1999)',
 'Total Recall (1990)',
 'Toy Story 2 (1999)',
 'Fugitive, The (1993)',
 'Men in Black (1997)',
 'Titanic (1997)',
 'Airplane! (1980)',
 'Babe (1995)',
 'Usual Suspects, The (1995)',
 'Galaxy Quest (1999)',
 'Sixth Sense, The (1999)',
 'Breakfast Club, The (1985)',
 'Blade Runner (1982)',
 'Election (1999)',
 'Groundhog Day (1993)',
 '2001: A Space Odyssey (1968)',
 'Casablanca (1942)',
 "Bug's Life, A (1998)",
 'Pulp Fiction (1994)',
 'Godfather: Part II, The (1974)',
 'Abyss, The (1989)',
 'Forrest Gump (1994)']
```

Fig: MAP Score for BERT4Rec

```
print("Mean Average Precision of BERT4Rec model = ", sum1/count1)

Mean Average Precision of BERT4Rec model =  0.2778572935105253
```

Fig. 10. BERT4Rec Results

Fig: Input is test dataset (Example user ID =270)

	userId	movieId	rating	timestamp	title
573101	270	1639	5	972669546	Man Who Would Be King, The (1975)
296897	270	1276	5	972669521	Maltese Falcon, The (1941)
215948	270	627	5	972669474	Mad Max (1979)
507133	270	384	5	972669159	Lethal Weapon (1987)
221497	270	955	5	968626656	Goldfinger (1964)
...	...	...	...	...	...
987784	270	3238	3	968622946	Adventures of Elmo in Grouchland, The (1999)
819337	270	1615	3	968622902	Aces: Iron Eagle III (1992)
975735	270	2057	3	968622839	52 Pick-Up (1986)
967469	270	2748	2	968625819	Exorcist II: The Heretic (1977)
948243	270	404	2	968625819	Exorcist III, The (1990)

90 rows × 5 columns

Fig: MAP Score for Hybrid model

```
print("Mean Average Precision for Hybrid model = ",sum1/count1)

Mean Average Precision for Hybrid model =  0.32803170385133684
```

Fig: Hybrid model output

```
['Braveheart (1995)',
 'Being John Malkovich (1999)',
 'Men in Black (1997)',
 'Fargo (1996)',
 'L.A. Confidential (1997)',
 'Princess Bride, The (1987)',
 'Shakespeare in Love (1998)',
 'Star Wars: Episode I - The Phantom Menace (1999)',
 'E.T. the Extra-Terrestrial (1982)',
 'Groundhog Day (1993)',
 'Alien (1979)',
 'Pulp Fiction (1994)',
 'Forrest Gump (1994)',
 'Terminator, The (1984)',
 'Fugitive, The (1993)',
 'Gladiator (2000)',
 'Ghostbusters (1984)',
 'Godfather: Part II, The (1974)',
 'Blade Runner (1982)',
 "Bug's Life, A (1998)",
 'Usual Suspects, The (1995)',
 'Abyss, The (1989)',
 'Aliens (1986)',
 'Galaxy Quest (1999)',
 'Total Recall (1990)',
 '2001: A Space Odyssey (1968)',
 'Babe (1995)',
 'Stand by Me (1986)',
 'GoodFellas (1990)',
 'Jaws (1975)']
```

Fig. 11. Hybrid model Results