HEAD OF DEPARTMENT

BSc Project Lab

Bikash Kumar Mahanti
candidate for **BSc** degree in **Computer Engineering**

Recommendation of movies using BERT

The purpose of this project is to exploring the BERT (Bidirectional Encoder Representations from Transformers) for movie recommendation. Here in this project, the MovieLens dataset (ml-100k) is used. The model is trained on MovieLens dataset in 3000 batches for 3 epochs and tested in 20% of the data from the dataset. The model utilizes the user behavior and movie features to provide personalization recommendations of the movies to the user.

Project overview

Below are key points that this project involves:

1. **Data Collection and Preprocessing:**
   - In this project we used MovieLens dataset (ml-100k).
   - u.data and u.item are the two files used from this MovieLens dataset (ml-100k).
2. **Model Development:**
   - Using the pre-trained BERT and customizing according to our needs for our dataset to understand better the relation between users and movies.
   - Adding extra elements to the model for better personalization for the user needs.
3. **Training and Evaluation:**
   - Training our model in pre-processed data for better performance and accuracy
   - And evaluating our model based on the 20% test data from the ordinal dataset to check up the necessary performance metrics score.
4. **Deployment and Recommendation:**
   - After the training, the model is deployed for use to recommend the movies for the user based on theirs previously watched movies and the genres they watched.
   - The recommendation function will return the top k-movies to the user based on his previously watched movies.

Deliverables:

This project delivers the following things:

- A movie recommendation function for personalized recommendation of movies based on user history
- A fully trained Model for movie recommendation
- Detailed documentation of the project
- Evaluation metrics and model architecture and example usage of the project.

Expected Outcome:

In this project, the aim is to enhance and optimize the BERT for personalized movies recommendation to the user based on his previously watched movies. And using the movie feature like genres along with the user data, exploring the capabilities of the BERT to providing insights and development in movies recommendation system.

Supervisor at the department: **László György Grad-Gyenge**

Budapest, 05.26.2024

MŰEGYETEM 1782

University of Technology and Economics of Budapest
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

Bikash Kumar Mahanti

RECOMMENDATION OF MOVIES USING BERT

SUPERVISOR

Grad-Gyenge László

György

BUDAPEST, 2024

Contents

Summary

This project focus is to develop and train a model for movie recommendations using BERT (Bidirectional Encoder Representations from Transformers) to enhance the user experience and providing a personalization recommendation to the user, based on his previously watched movies. The aim is to effectively analyze user preference and using them with movies features such as genres to give a better recommendation of movies.

The project begins with the data loading and preprocessing of the data. In this project we are using the MovieLens dataset (ml-100k). it's an open-source dataset available in Kaggle or MovieLens website. More specifically, we will use the u.data (for the ratings information) and u.item (for the movie information). The details about the type of information and the schema of the dataset can be found inside README file.

To integrate this dataset into our recommendation system, we need to first preprocess this data. Therefore, the genres are encoded as binary features. This helps our information into a more suitable format for our model. Then a BERT tokenizer is employed to preprocess our data by converting them into tokenized inputs that BERT can process and understand them.

A custom PyTorch dataset class called **'MovieLensDataset'** is created to handle the pre-processed data. This class prepares the dataset for the model to give as an input, including the tokenized the movie titles and genre features, and then pairs them with the corresponding user ratings. Also, it creates a data loader to facilitate the data loading in batches for more efficient model training.

The core of this project is the model development. In this project, the **'BertWithGenreFeatures'** model is used. This model integrates, BERT to process movie titles (textual data) and combines the output with genre features to predict user ratings. This model architecture includes linear layer that processes the BERT outputs and the genre features for the movie ratings predictions.

After the model development, the next step is to optimize the hyperparameters for this model to train effectively and minimize the prediction error. The model is trained using the *AdamW* optimizer, with a learning rate to adjust the learning rate dynamically. The training loop iterates over multiple epochs and updating the model weights and monitoring the training loss.

Once the model is trained, then the model is deployed for recommendation of the movies. A recommendation function **'recommend_movies'** is created for this purpose. It is implemented to predict user ratings for the unwatched movies and then sort them in descending order. After that, it returns the top-K movies for that user Id.

The project's outcome is a movie recommendation system that leverage BERT a powerful text processing capabilities and then integrates additional features like genre features to make it more personalization recommendations of movies to the user.

Introduction

In today's time of too much information, individualized suggestion tools are very important in many areas, especially in entertainment. As the number of things to watch continues to grow, people depend more on these suggestion tools to find things they like. A big use of these tools is in movie places online, where good suggestions can really make people happy and keep them interested. This plan is to make a better movie suggestion tool that uses BERT (Bidirectional Encoder Representations from Transformers) to give each person suggestions based on what they like and what each movie is like.

Motivation

The inspiration to work on this project arises from the ever-increasing demand for accurate, personalized recommendations in the entertainment sector. The existing recommendation algorithms fail to grasp the complex relation between the users and the items which leads to poor recommendations. A recent breakthrough in the field of natural language processing (NLP), the BERT model, can solve this problem quite efficiently. BERT is trained to handle text data in a contextual manner which proves to be a perfect fit for recommendations. We can feed the titles and genres of movies to BERT and use its encoding as a better representation of the movies. This will ultimately lead to more relevant and accurate recommendations.

Objectives

Here, we will list some major tasks needed to finish in this project. This project aims to build a state-of-the-art movie recommendation system using BERT. As a beginning, we will focus on collecting and preprocessing the data including both movies and users. The dataset used here is from MovieLens which is publicly available. Some statistics and basic descriptions of the dataset can be accessed here. We will go through cleaning and encoding the data to feed into our recommendation model. This step is very important to ensure the quality of the data that feeds into the model which in turn effects the final recommendation results. After data preprocessing, we will move on to develop and train the recommendation model. Only with a well-prepared dataset and well-trained model, we finally generate accurate recommendation results for the movies.

After data preprocessing, we will focus on model development. A deep learning model based on BERT will be created and implemented from scratch. Given that the input to the recommendation model includes text data such as movie titles and genre, we will investigate how to successfully use BERT for such text data to provide meaningful and useful representations or embeddings. We also construct unique dataset classes and data loaders to improve the model's training efficiency. With this, our recommendation model may be easily implemented into a real application system. Finally, we assess the recommendation model on a test set and visualize the results to demonstrate the system's performance. After achieving all of these objectives, we will have a cutting-edge movie recommendation system using BERT that can successfully propose accuracy.

Data loading and preprocessing

Please read below for the detailed information about the data preprocessing and data loading.

Step 1: Read and pre-process Movie Data

The process starts by fetching movie data, including important attributes like MovieID, Title, and genre classification. Using Pandas, movie data is read from a CSV file into a DataFrame, and associated with specified column names and data types. Besides, to simplify operations and focus on context, other attributes like ReleaseDate, VideoReleaseDate, IMDbURL etc. are dropped from the DataFrame This preprocessing step the aim is to refine the dataset, remove noise and increase its suitability for further analysis.

Step 2: Read the User-Item Ratings Data

At the same time, user-item rating data is retrieved, which contains important information including UserID, MovieID, and corresponding ratings given by users. By using the Panda function, the ratings data is exported to a DataFrame from a tab-separated file, associated with a defined column name and data structure. This step ensures that the user's preferences are seamlessly incorporated into the recommendation process, making it easier to create personalized recommendations for individual users.

Step 3: Integrate Movie and Ratings Data

Connecting movie and rating data is an important step in the data creation pipeline. The generic feature of MovieID creates a unified data set by combining the two data sets, combining movie features and user ratings. This combined dataset forms the basis for subsequent research, which enables the investigation of the relationship between movie features and consumer preference.

Step 4: Encode Genres as Binary Features

To facilitate exemplar training and enhance his understanding of films, categories of text are included as two components. This transformer modifies the classification of movies in accordance with a deep learning algorithm, enabling the inclusion of attribute information in the proposal model the resulting dataset includes a diverse set of film elements, with two sets of attributes parts are included, ready for use in model training and analysis

Step 5: Use the PyTorch Dataset Class

In preparation for model training, a custom PyTorch dataset class, 'MovieLensDataset', has been created to hold the processed data. This class makes it easy to create input samples for the model, including tokenized movie titles, attention masks, genre binary features, and associated ratings by providing data in a PyTorch-compatible format, the dataset class generates data integration is easy into training pipelines, ensuring alignment with deep learning processes

Step 6: Create Data Loaders

Finally, the data loader is instantiated to enable efficient batch processing during model training. Using PyTorch's Data Loader module, processed data is divided into groups of specified sizes, which simplifies parallel computation and optimizes memory usage The resulting data loaders act as conduits for data into the recommendation model seamlessly, providing robust and scalable training algorithms.

Model Development and Training

Step 7: Define the Transformer Model by additional attributes

At the core of the recommendation process is a transformer model with additional features to implement the capabilities of BERT (Bidirectional Encoder Representations from Transformers), a custom program called **'BertWithGenreFeatures'** has been developed. This model design incorporates genre features through a combination of BERT transformer and complementary layers, increasing model sensitivity to movie properties on textual data by combining BERT-encoded representations with genre features and treating them as a linear layer on the inside, the model learns to impose user evaluations based on broad.

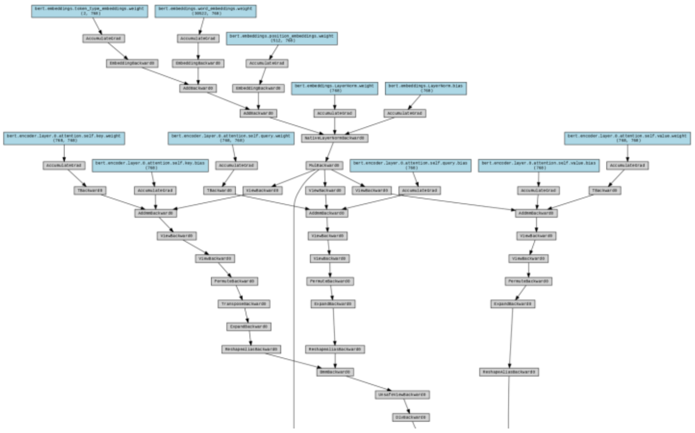
Step 8: Transformer model training

Once the model architecture is defined, the training phase begins, aiming to optimize the model parameters to minimize prediction errors. The transformer model, modeled using the defined architecture, is trained using a combination of methods, such as gradient descent optimization and learning rate scheduling. The AdamW optimizer is used to update the model parameters, while the schedule sequence and warmup ensure the smooth running of the training throughout the training process. The model is trained over multiple periods, iteratively through sets of data extracted from preprocessed data sets.

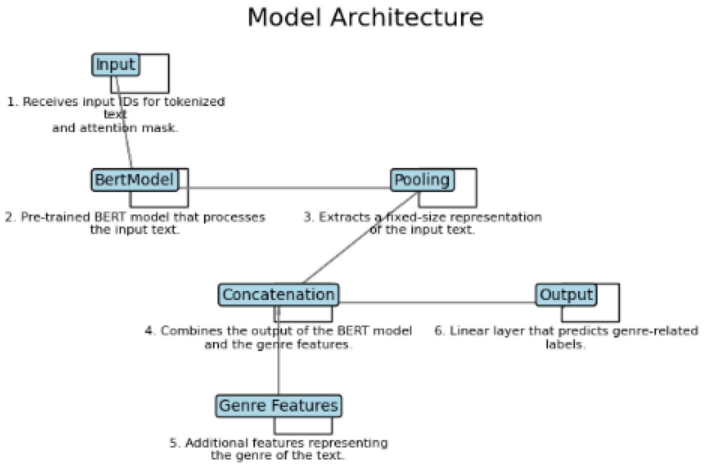
Training Loop

Within each epoch, the training loop iterates over batches of data, feeding them into the model for forward pass computations. The loss function, defined as the Mean Squared Error (MSE) loss, quantifies the disparity between predicted and actual user ratings. Backpropagation is then employed to compute gradients and update the model parameters, facilitating the convergence of the model towards optimal performance. Additionally, gradient clipping is applied to prevent exploding gradients and stabilize the training process. Progress is monitored throughout training, with periodic updates on the running loss to assess model performance and convergence.

Detailed model view



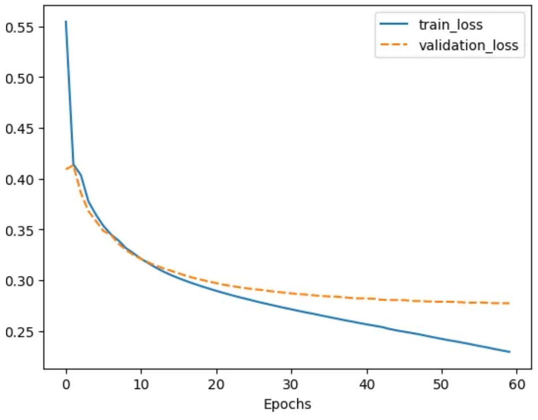
Abstract model view


Model Architecture
Input
1. Receives input IDs for tokenized text and attention mask.
BertModel
Pooling
2. Pre-trained BERT model that processes the input text.
3. Extracts a fixed-size representation of the input text.
Concatenation
Output
4. Combines the output of the BERT model and the genre features.
6. Linear layer that predicts genre-related labels.
Genre Features
5. Additional features representing the genre of the text.

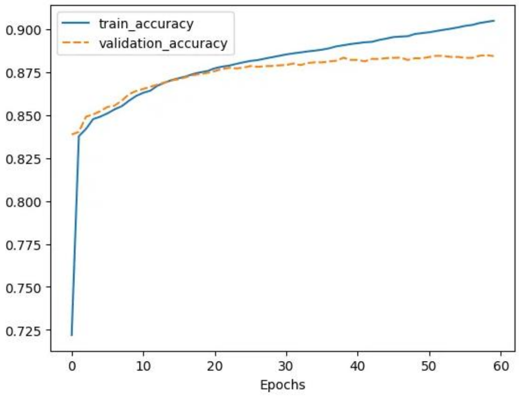
Outcome and Evaluation

Upon completion of training, the model is primed to generate accurate and personalized movie recommendations based on user preferences. Evaluation metrics such as training loss and validation metrics may be utilized to assess the model's efficacy and generalization capabilities. Through meticulous model development and training, the recommendation system is poised to deliver superior user experiences, providing tailored movie suggestions that cater to individual tastes and preferences.

By meticulously crafting the model architecture and orchestrating its training process, the recommendation system endeavors to fulfill its objective of delivering accurate, personalized, and engaging movie recommendations, thereby enhancing user satisfaction and engagement in movie consumption experiences.



Loss over epoch visualization

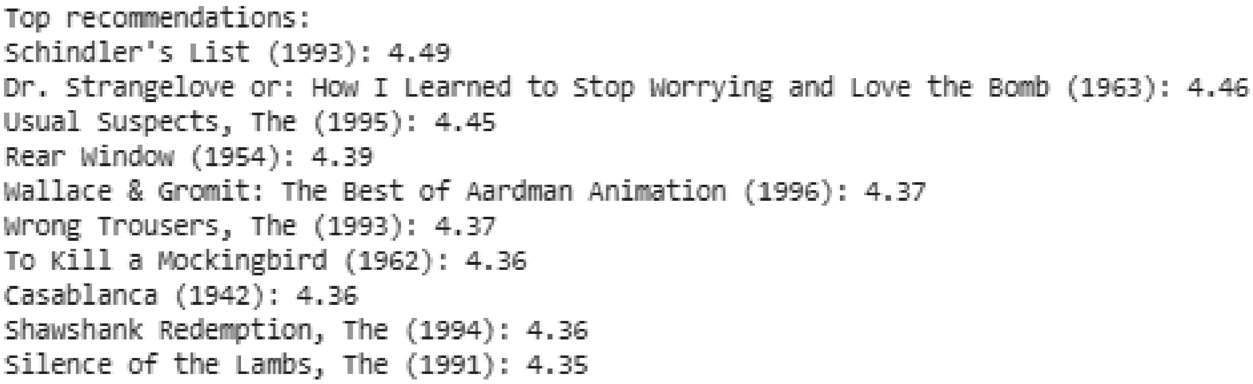


Accuracy over epoch visualization

recommend_movies function

The recommend_movies function is an integral part of the movie recommendation system, and allows you to create personalized movie recommendations based on user preferences. Working with parameters like user ID, trained model, tokenizer, movie data, data frames including all movies, this function extracts movies already watched by the user, gives suggestions for movies they have not watched Leveraging BERT-based model predictions, tokenizers with genre features Together with movie titles tokenized by , the project calculates the predicted numbers for the candidate movies and then ranks these predictions based on their ratings, returning K recommendations with top to the user. By carefully combining deep learning techniques with data preprocessing, the project empowers recommendation systems to deliver customized movie recommendations, satisfying users and their engagement with film consumption experiences is greater.

Output



```
Top recommendations:
Schindler's List (1993): 4.49
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1963): 4.46
Usual Suspects, The (1995): 4.45
Rear Window (1954): 4.39
Wallace & Gromit: The Best of Aardman Animation (1996): 4.37
Wrong Trousers, The (1993): 4.37
To Kill a Mockingbird (1962): 4.36
Casablanca (1942): 4.36
Shawshank Redemption, The (1994): 4.36
Silence of the Lambs, The (1991): 4.35
```