CSE 6240: Web Search & Text Mining

TransRecG: Transformer-Based Recommendation System using Graph Embeddings

Sai Prasath Suresh, Ishwarya Sivakumar, Jeongjin Park, Balaram Behera

Introduction

INTRODUCTION

Sequential Recommendation Task

PROBLEM DESCRIPTION

Given a user U, and the user's **item history**, predict the rating the user will give to the next item. We consider the movie-rating prediction task for this project.

MOTIVATION / INTRODUCTION

Improving Deep Learning-based Recommendation Systems (DLRS)

PROBLEM WITH EXISTING DLRS

- Ignore the **temporal information**
- Fail to take advantage of the **higher order connectivity features** like **user-user and item-item relations.**

IMPORTANCE OF PROBLEM

- Recommendation System that is **static** and **does not utilize higher-order connectivty** won't be able to perform well when user's preference changes dynamically, which is often the case in real life.
- Using higher-order connectivity information, relationships between movies, and users can be learnt to provide **customized and accurate** recommendations.
- Can impact many industry-based recommendation systems like Netflix (movies), Amazon (products), and Spotify (songs).

Our Approach

Creating a Novel Model!

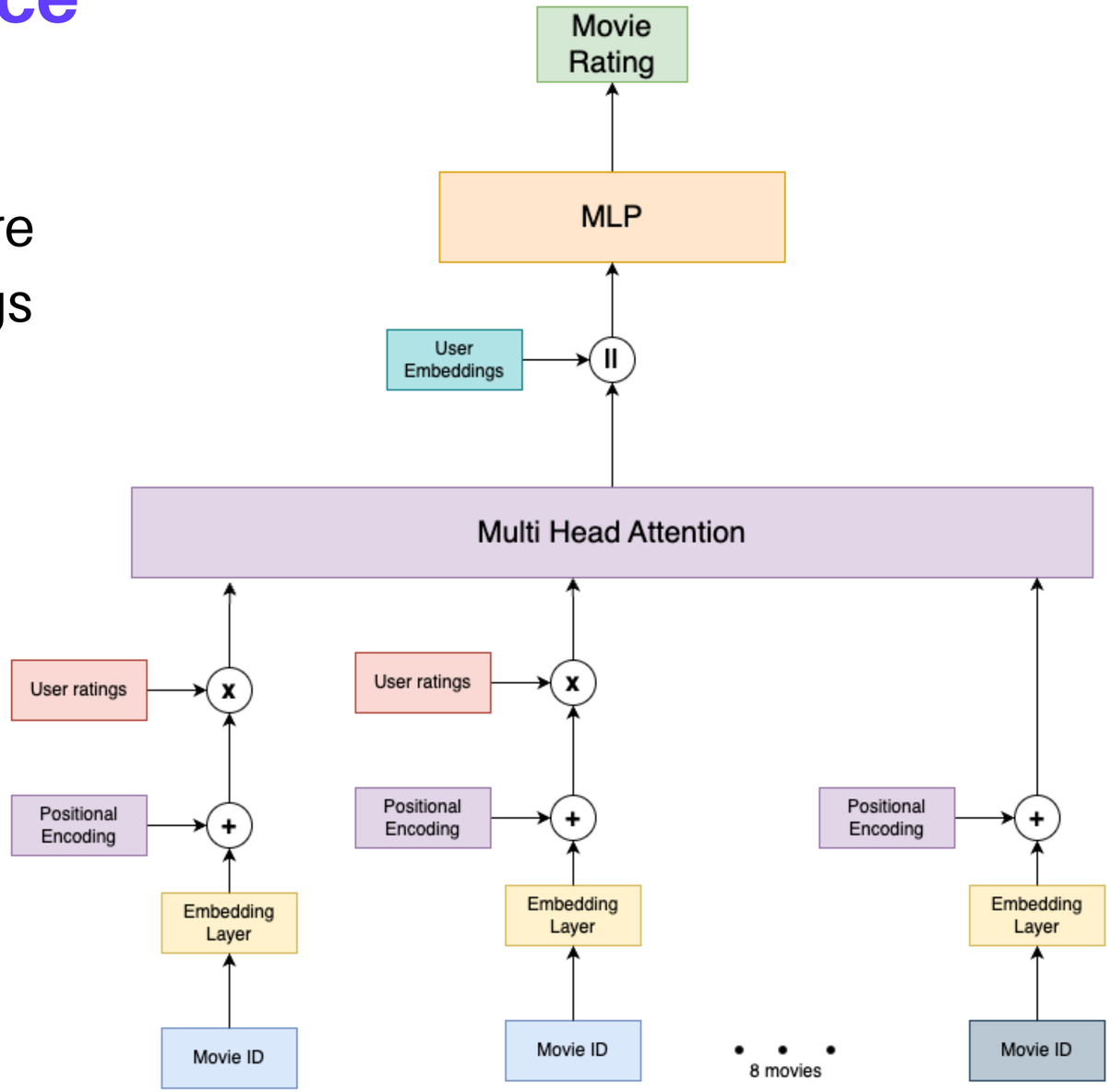
Architecture of Baselines

Baseline 1: Behaviour Sequence Transformers (BST)

- Transformer Encoder-based architecture
- No pre-trained embedding. Embeddings are generated using user and movie metadata.
- Captures user embeddings, and sequential ordering of movies.

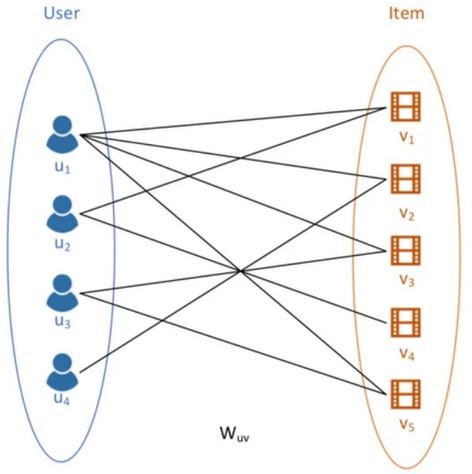
Model Description:

- Uses Multi-Head Attention = 9 Heads
- Embeddings Size = 63
- 4 Layer MLP (589, 1024, 512, 256, 1)



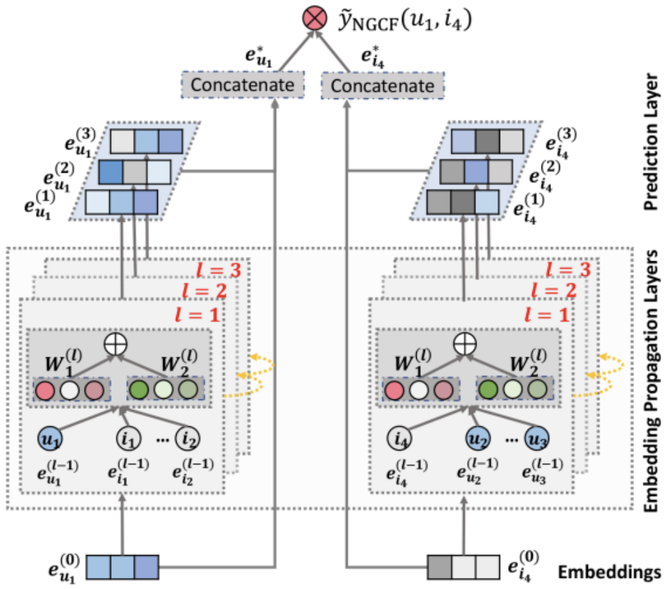
Baseline 2: Neural Graph Collaborative Filtering (NGCF)

- User-Movie Bipartite graph. Edge exist if the user has rated the movie. (Rating is ignored)
- Nodes initialized with adj matrix representation.
- GCNConv used for generating node embeddings on the rating prediction task.



Model Description:

- Number of Nodes = 9993
- Number of Edges = 800167 (train) + 100020 (val) + 100022 (test) = 1000209
- 2 Layer Message Passing (9993, 18, 18)



Limitations of Existing Baselines

BEHAVIORAL SEQUENCE TRANSFORMER (BST)

- Uses Transformers to capture the sequential ordering of movies
- The recommendations are made personalized using user embeddings
- **Cons**:
  - Does not capture higher order user-user or item-item relations as it considers every user as an independent entity and learns their embeddings individually.

NEURAL GRAPH COLLABORATIVE FILTERING (NGCF)

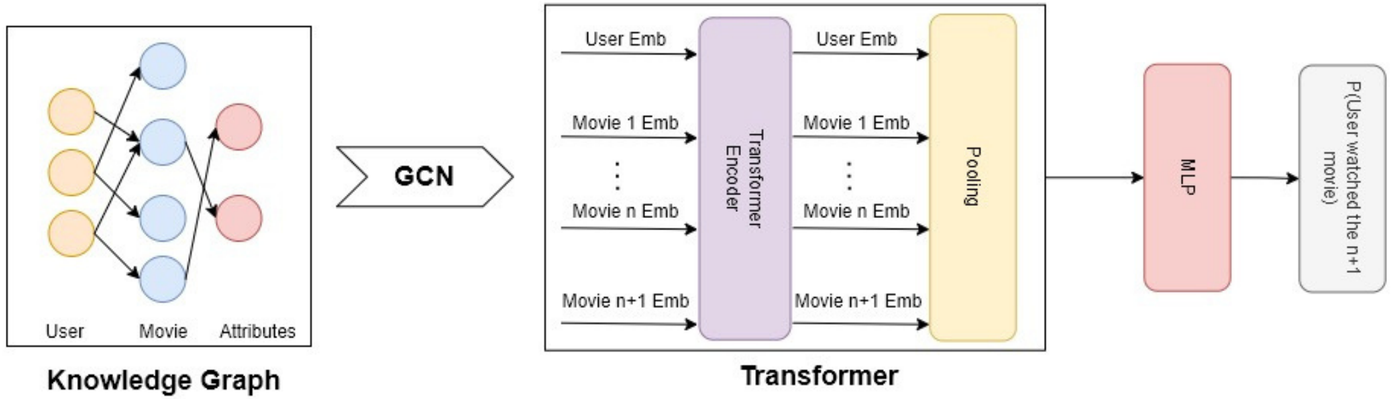
- Exploit the user-item graph structure by propagating the embeddings on the graph and performing the edge prediction on the generated embeddings.
- Expressive modeling of the higher order connectivity features in the user-item graph.
- **Cons**:
  - Does not consider the temporal behavior of the user to make recommendations.
  - Doesn't model the underlying user-user or item-item relationships.

Architecture of TransRecG

Our Model: TransRecG

GOAL

Build a recommendation system (RecSys) that uses BST and NGCF on a User-Movie Knowledge Graph (KG)



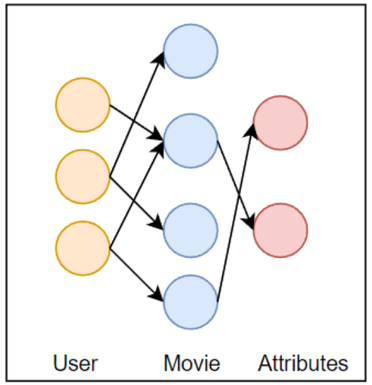
HOW DOES IT WORK?

- BST captures the sequential information to understand the temporal behavior of the user.
- NGCF model on the Movie-User KG captures the user-user, movie-movie and user-movie relations.
- Augmenting the Transformer with the RGCN embeddings from the KG helps to leverage temporal and spatial information to make better and personalized recommendations

NOVELTY

- Using a BST with a NGCF model to build a RecSys that captures both temporal information and also exploits the underlying relations like user-user, user-movie and movie-movie.

Part 1: NGCF on Knowledge Graphs (KG)



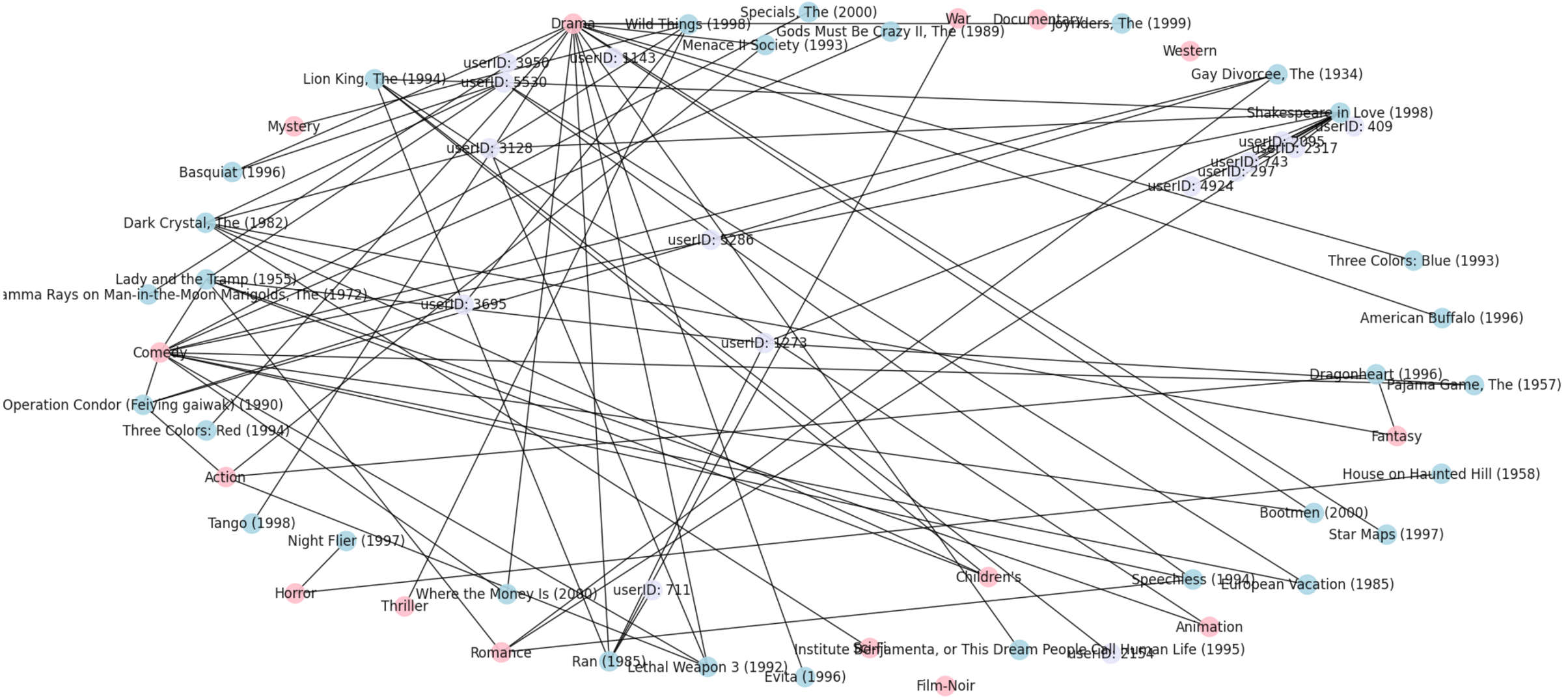
Knowledge Graph

- User-Movie-Attribute Knowledge graph.
- Attribute = {Genre}
- Movie and Attribute initialized with GloVE embeddings.
- User node embeddings are randomly initialized.
- User-Movie edges are defined by the ratings - 5 types (1, 2, 3, 4, 5)
- Movie-Attribute edges all belong to the same type - (6)
- **RGCN** is used for generating the node embeddings, trained on the rating prediction task.

Model Description:

- Number of Nodes = 100010
- Number of Edges = 1000209 (User-Movie) + 6408 (Movie - Attribute) = 1006617
- 2 Layer Message Passing (50, 50, 50)

User - Movie - Attribute KG

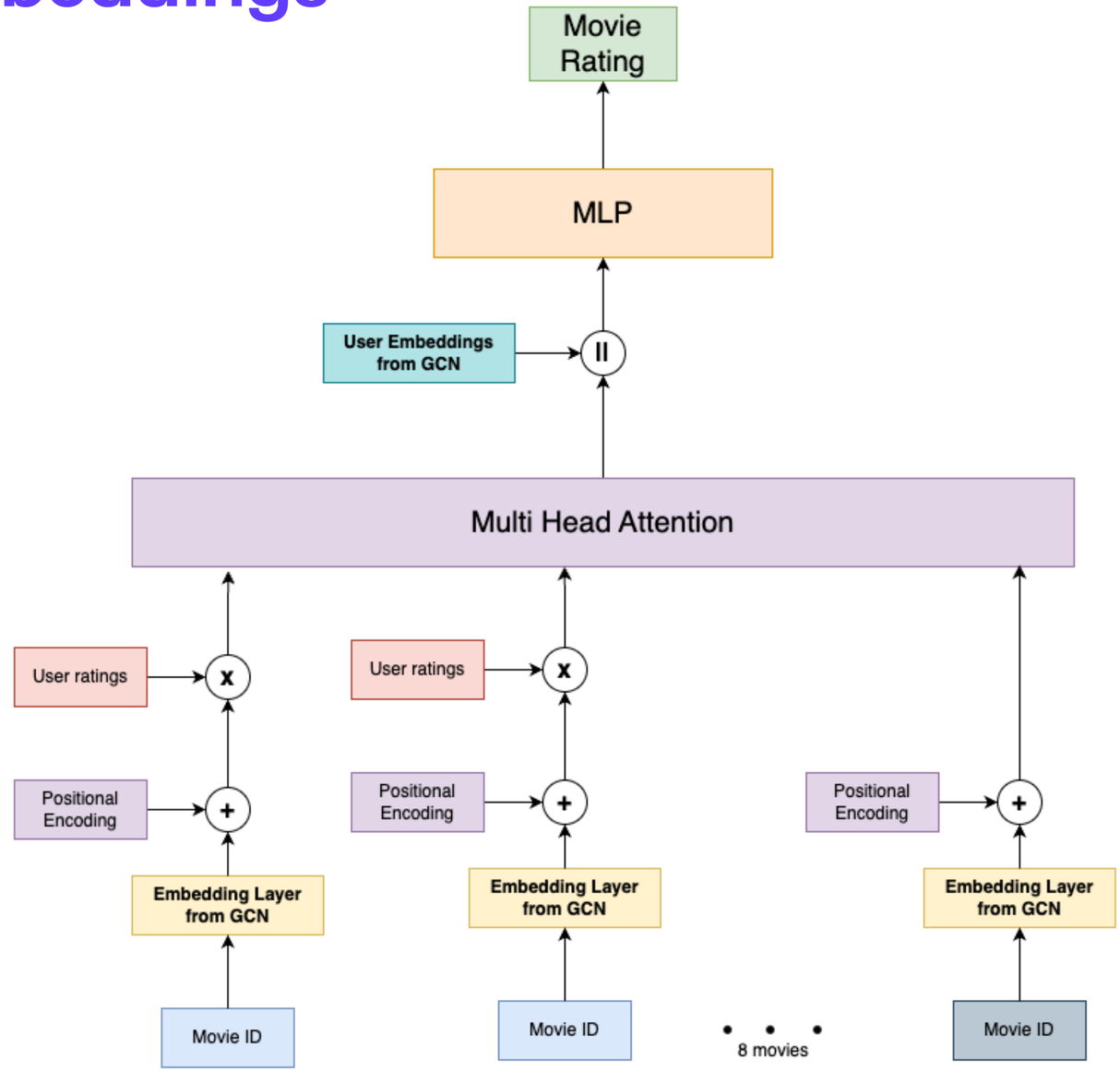


Part 2: BST with NGCF (+KG) Embeddings

- Augmented BST with graph embeddings for movies and users.
  - User/Movie embeddings from User-Movie bipartite graph (BP).
  - User/Movie embeddings from User-Movie-Attribute KG.
- Concatenate the graph embeddings with User/Movie embeddings before passing to the Transformer.

Model Description:

- Movie/User Embedding: Representation at the end of 2nd Message Passing Layer
- BP Embedding = 18, Attention Heads = 9
- KG Embedding = 50, Attention Heads = 15
- 4 Layer MLP



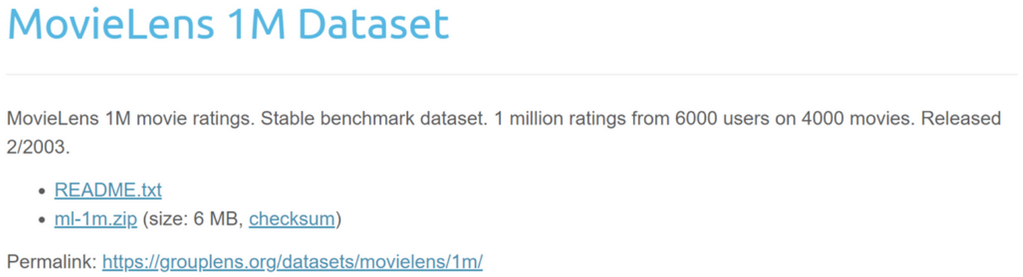
Experiment Setup

Data, Experiments and Evaluations.

Data

HOW WAS DATA OBTAINED ?

- Used the publicly available ***MovieLens-1M*** **dataset.**
- It has 3 files
  - users.dat: metadata about the user.
  - movies.dat: metadata about the movies.
  - ratings.data: (user, movie, rating, timestamp)



MovieLens 1M Dataset

MovieLens 1M movie ratings. Stable benchmark dataset. 1 million ratings from 6000 users on 4000 movies. Released 2/2003.

- README.txt
- ml-1m.zip (size: 6 MB, checksum)

Permalink: https://grouplens.org/datasets/movielens/1m/

DATA PREPARATION

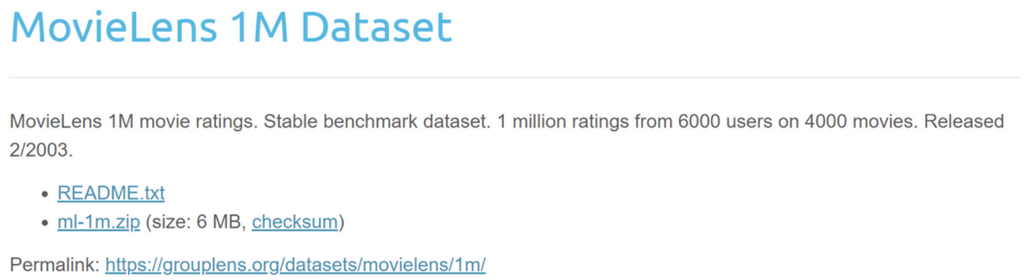
- Split into train, test and validation test based on the **timestamp**. The split was 80-10-10.
- Sequence size of 8 was chosen ----> For a user, given 7 movies and their corresponding ratings the model will predict the rating given by the user for the 8th movie.
- Since we consider only the ratings given by a user when making a recommendation for a user, normalization of ratings was not done.

https://grouplens.org/datasets/movielens/1m/

Data

PROPERTIES OF DATA

- 6,040 users and 3,883 movies.
- **Each user has given reviews for at least 20 movies**
- Average number of ratings per user is 164.7
- Average rating is 3.58
- **611 users in the validation set and 200 users in the test set do not having data in the training set**
- With a sequence size of 8, there are
  - 757077 ratings in the training set
  - 92113 ratings in the validation set
  - 91185 ratings in the test set
- Drama is the most popular genre



MovieLens 1M Dataset

MovieLens 1M movie ratings. Stable benchmark dataset. 1 million ratings from 6000 users on 4000 movies. Released 2/2003.

- README.txt
- ml-1m.zip (size: 6 MB, checksum)

Permalink: https://grouplens.org/datasets/movielens/1m/

https://grouplens.org/datasets/movielens/1m/

Experimental Setup

- **Loss Function:** Root Mean Squared Error (RMSE) Loss
  - torch.nn.MSELoss()
- **Optimized:** Adam Optimizer
  - torch.optim.AdamW()
- **Batch Size, Learning Rate, Number of Epochs:** Optimized separately for each model

Results

Comparing our results with baselines.

Evaluation Criteria

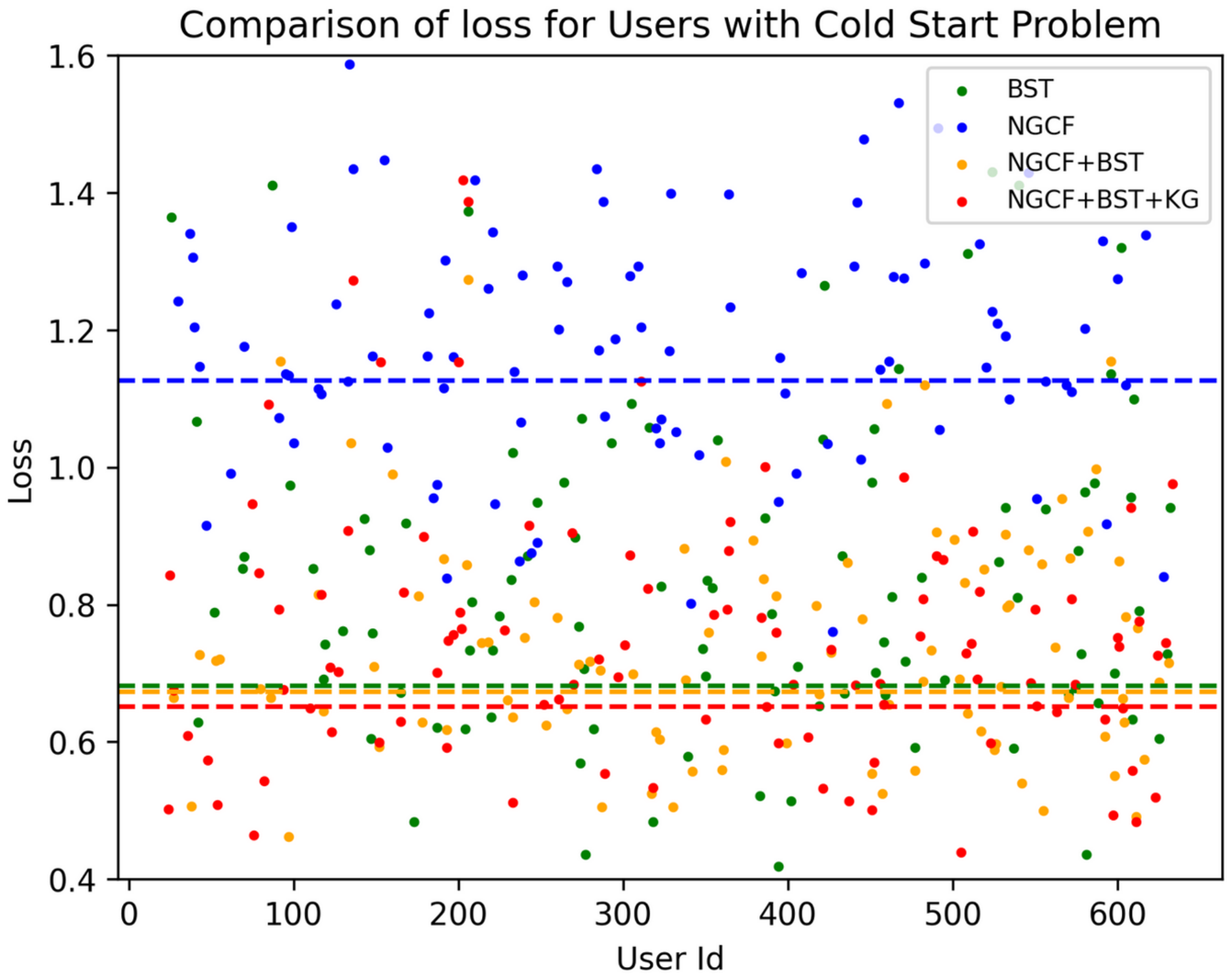
- The **Mean Absolute Error** (**MAE**) **Loss** for the rating prediction task was computed across various models.
- Performance of different models were compared and analysed, to understand their strengths and weaknesses.
- Explored the ability of the model to handle the **user cold start problem.**
- Analyzed the tradeoff between number of training samples and loss.

Model	Model Description	Loss
GRU	No pre-trained embedding. Sequence size 8.	0.792
BST - 2	No embeddings from GCN. Sequence Size 2	0.786
BST - 4	No embeddings from GCN. Sequence Size 4	0.767
BST - 8	No embeddings from GCN. Sequence Size 8	0.760
BST - 15	No embeddings from GCN. Sequence Size 15	0.754
NGCF - BP	GCN on User-Movie bipartite graph	1.148
NGCF - KG	RGCN on User-Movie KG	1.095
BST - 8 + NGCF - BP	Embeddings from the NGCF on Bipartite Graph	0.746
BST - 8+ NGCF - KG	Embeddings from the NGCF on User-Movie KG	0.741

Results

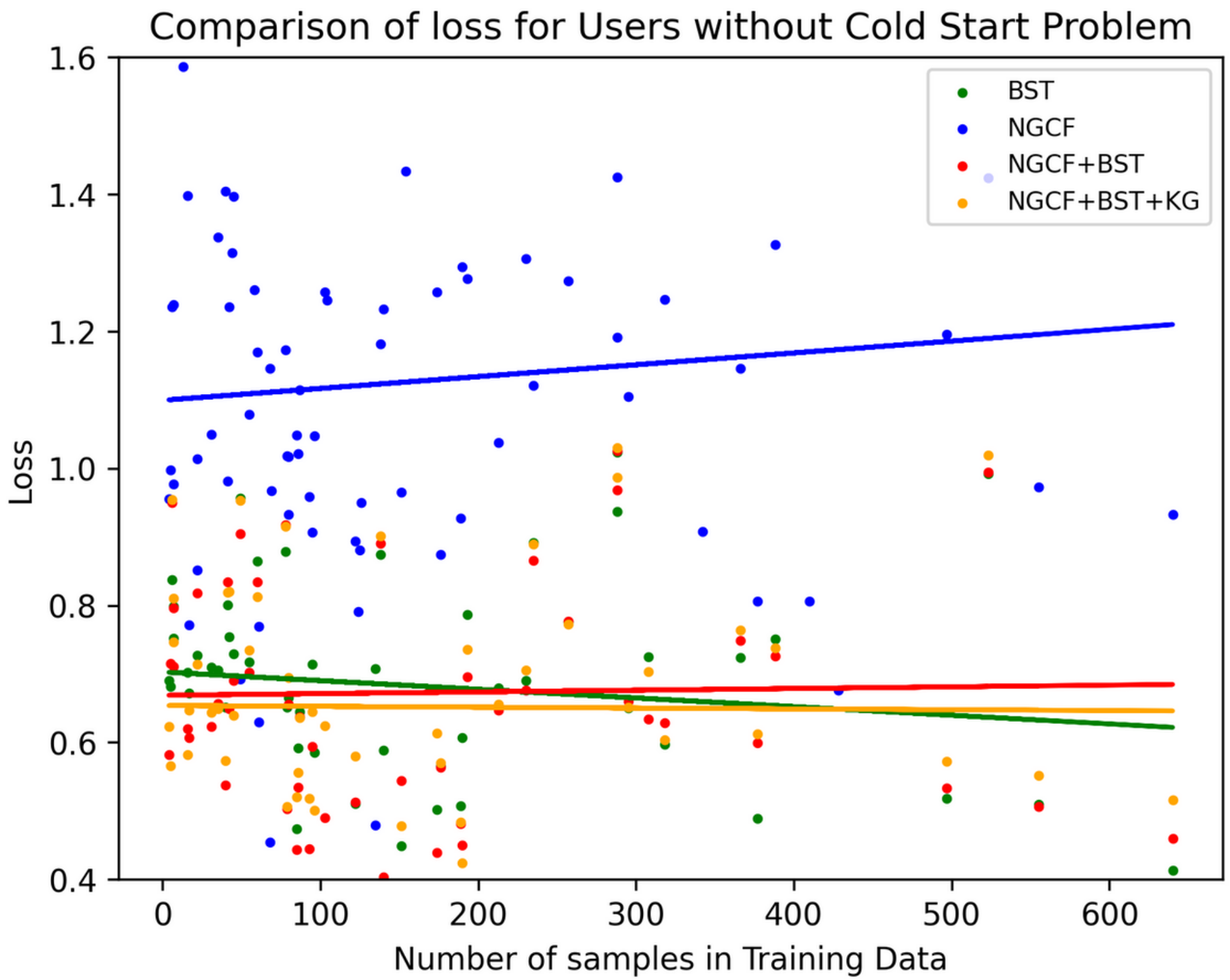
- Sequence based models (GRU and BST) perform better than non-sequence based models (NGCF).
- Longer Sequences lead to better performance.
- Augmenting BST models with graph embeddings improves the performance of the model.
- Graph embeddings from RGCN + KG performs better than GCN + BP models, as initial embeddings are more meaningful (GloVE).

Cold Start Problem



- NGCF performs the worst, as no information about the user is available.
- The sequence based models are able to overcome this problem by analysing the sequence of previous movies watched by the user (given as input), and recommending relevant movies.
- Adding graph embeddings to BST reduces the loss, as movie-movie relations are learnt through 2-layer message passing.
- However, KG achieves the least loss as good movie embeddings (through attribute edges) are generated by the KG.

Number of Samples vs Loss



- The loss for NGCF increases, with more samples per user. If the user's preferences are varied and dynamic, NGCF does not perform well.
- Comparatively, the loss in BST reduces with more samples as its able to capture the dynamic preferences of the user.
- However, the performance of BST models augmented with graph embeddings almost remain unchanged.
    - We believe that adding an attention based weights to balance between graph and meta data embeddings, can improve the performance of these models

Future Works

How to improve the TransRecG Model?

Future Work

- The BST model uses a sequence length = 8 (due to limited compute). The effect of sequence length on the model performance should be further analysed.

- We implement the Knowledge Graph (KG) as a heterogeneous graph only in terms of the edges. Creating the KG as a fully heterogeneous graph with both different types of nodes and edges, is a relevant future work as user-movie-attribute graphs are composed of different types of entities in nature.
    - Better user embeddings can be generated by using user meta data instead of random initialization.

Thank You!