**TransRecG: Transformer-Based Recommendation Systems using GCN Embeddings**

**[CSE 6240: Web Search and Text Mining - Team 6 ]**

**Summary:** The TransRecG model solves the sequential recommendation task in the context of movie rating prediction. The model learns higher order user-user, and movie-movie relations using the Relational Graph Convolutional Network model on a Knowledge Graph. The learnt embeddings are then used by the Behavioural Sequence Transformer model, for predicting the rating given the user, and the previous movies rated by the user.

**Folders:**

1. data: Contains all the raw and processed data for the project.
2. weights: Stores the optimized parameter weights for all the models used in this project.
3. misc\_files: Temp files created for data and model analysis.
4. notebooks: Contains the colab notebooks that implement the data preprocessing, model training, and final analysis.

**data:**

The data can be downloaded from the link: <https://drive.google.com/drive/folders/17km6cX3P3mPt33SLgp2I3qYBNMhhqkJk?usp=sharing>.

* ml-1m.zip: Contains the raw data used in this project.
  + users.dat: Contains the user\_id, sex, age\_group, occupation and zip code.
  + movies.dat: Contains the movie\_id, title and genres (multi-valued attribute)
  + ratings.dat: Contains user\_id, movie\_id, rating and unix\_timestamp.
* movies.csv, ratings.csv, users.csv: Contains processed movie, rating, and user information. (Categorical features are one-hot encoded)
* train.csv, val.csv, test.csv - Contains the train-val-test dataset used for this project. The dataset is split using time stamps of the rating, with a 80-10-10 split.

**misc\_files:**

* train\_user\_count.pkl, val\_user\_count.pkl, test\_user\_count.pkl: Stores the count of data points available for each user in the train, val and test datasets.
* train\_user\_loss.pkl, val\_user\_loss.pkl, test\_user\_loss.pkl: Stores the mean MAE Loss of users in train, val and test dataset.

**notebooks:**

* Data Preprocessing.ipynb: Contains the code for loading the raw data, processing the data, and creating the train-val-test dataset.
* GRU implementation.ipynb: Implements a One layer Unidirectional GRU for the rating prediction task.
* BST.ipynb: Implements the Behaviour Sequence Transformer model for the rating prediction task.
* NGCF: Implements the Neural Graph Collaborative Filtering on the User-Movie Bipartite graph to obtain user and movie embeddings.
* KG\_GloVE\_Emb.ipynb: Creates the User-Movie-Attribute Knowledge Graph and initializes the node embeddings using GloVE embeddings.
* KG\_RGCN: Implements the Relational Graph Convolution Network model on the KG to obtain user and movie embeddings.
* NGCF\_BST: Implements the TransRecG (proposed model) by combining the embeddings obtained from the graph neural networks (both GCN on bipartite graph and RGCN on the KG), with the BST model.
* analysis.ipynb: Analyzes the MAE Loss of various models on the test dataset, compares the performance of models on handling cold start users, and studies the tradeoff between number of training dataset samples, and test loss.

**All our code files are in .ipynb notebook format, and all required files and libraries are loaded in the notebook itself. All loss functions, optimizers and parameters used by the models are defined in the codebook.**

**To execute the whole project, run the notebook in the order in which the codebooks are listed.**

**weights:**

* **BST:**
  + bst\_noemb.pkl, bst\_9\_head\_noemb.pkl: BST model weights with different number of attention heads.
* **GCN + BP (NGCF):** 
  + ngcf.pth, ngcf\_latest.pkl, ngcf\_model2.pth - Final GCN model weights after training.
* ngcf\_emb.pkl, ngcf\_emb\_18.pkl - Final user and movie embeddings obtained from the GCN model trained on the bipartite graph.
* **RGCN + KG:**
* genre\_emb.pkl - Initial embeddings of the attribute (genre) nodes in the KG obtained from GloVE embeddings.
* movie\_emb.pkl - Initial embeddings of the movie nodes in the KG obtained from GloVE embeddings.
* final\_kg\_weights.bst - Weights of the RGCN model trained on the KG.
* kg\_emb.pkl - Final user and movie embeddings obtained from the RGCN model trained on the KG.
* **NGCF + BST:** 
  + ngcf\_bst.pth: Final weights of the BST model which uses the user and movie embeddings from the GCN + BP model.
* **RGCN + KG + BST:** 
  + ngcf\_bst\_kg.pth, ngcf\_bst\_kg2.pth: Final weights of the BST model which uses the user and movie embeddings from the RGCN + KG model.