User-based Ensemble Game Retrieval System

Tianyi Ge, Dongjian Chen, Shukai Fan
[1] School of Information, University of Michigan

Introduction

This project build a user-based ensemble game retrieval system based on traditional searching model and recommendation models. In our project, we design an ensemble recommendation model that combines memory-based collaborative filter model with machine learning model. We combine our recommendation model with seraching model to construct our final retrieval system.

Objectives

1) Build a classical searching model
2) Build a recommendation model combining momery-based collaborative filter model with machine model.
3) Ensemble searching model and recommendation model.
4) Design evaluation experiment for our final model.

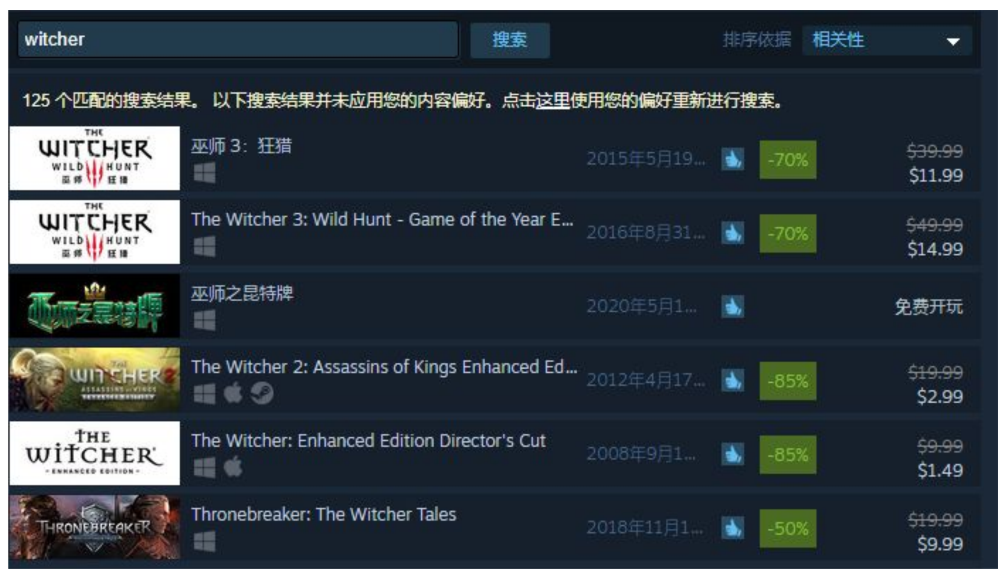

Figure 1. Steam searching engine

Design and Implementation

Dataset:

1) We get steam game data set from kaggle website:
https://www.kaggle.com/nikdavis/steam-store-games?select=steam.csv
This dataset is in csv format and contains lots of information of games on steam platform such as genre, publisher, language, etc.

2) We get our user-behavior and user-item data set from cseweb website:
https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data
This dataset is in JSON format and contains lots of user data including playtime, reviews, etc.

System overview:

The overview of our ensemble retrieval system.

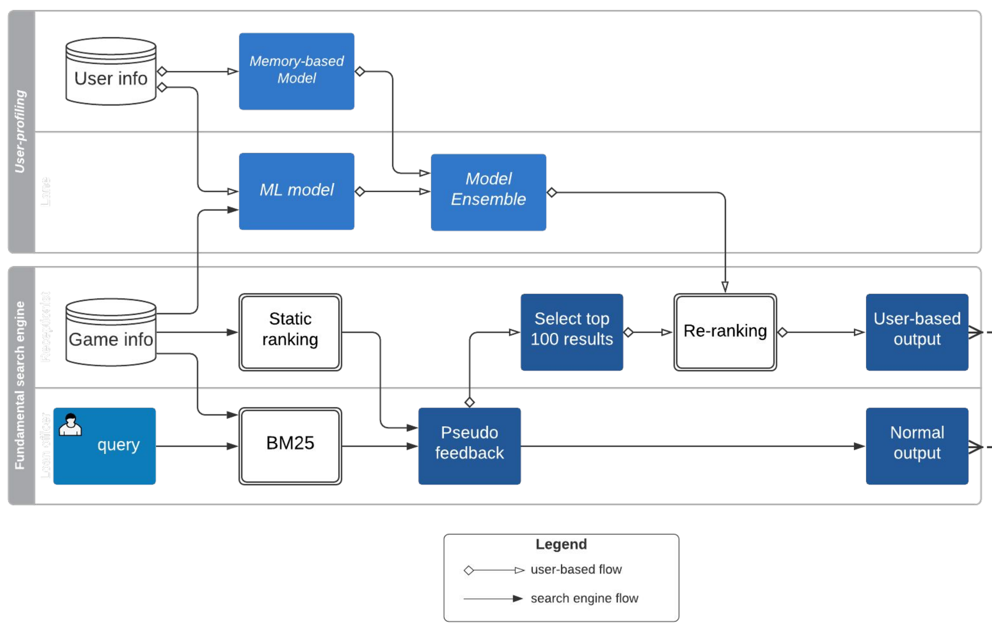

Figure 2. Ensemble retrieval system overview

Model design and implementation:

1) Search model:

We use a classical searching model based on document-query matching score BM25. The top 100 query results are retrieved as candidates to be re-ranked by integrating ensemble model metrics.

2) Recommendation model:

Memory-based model:

This model explores user playtime per game, and if a user recommends a game. This model benchmark three collaborative filtering techniques by 5-fold cross-validation on MAE. For the log-scaled data, SVD outperforms the other two to give a better approximation.

Rank	Algorithm	MAE
1	SVD	1.639161
2	KNNWithMeans	1.726102
3	Co-clustering	1.841283

Figure 3. Model evaluation

Machine learning model:

We choose four models and evaluate their performances to choose the best model in our recommendation task. The final model we choose is XGboost as shown in Figure 3.

	Model	MAE	NDCG
0	MLP Regressor	1.784809	0.957221
1	SVR	1.890197	0.951428
2	RandomForest	1.738922	0.954196
3	XGBoost Regressor	1.723970	0.957758

Figure 4. Model evaluation

After selecting the XGBoost as final model ,we use grid search and cross validation to do further parameter tuning.

Evaluation

	Method	Average Rank
1	Pure Retrieval	5.23
2	with Memory-based Model	4.62
3	with Machine Learning Model	4.95
3	with Ensemble Model	4.41

Figure 5. Final rerank evaluation

We tested the four models for 100 sample user with generated Pseudo query and target game. The ensemble model significantly helps to improve the retrieval results and can discover user desired games more accurately.

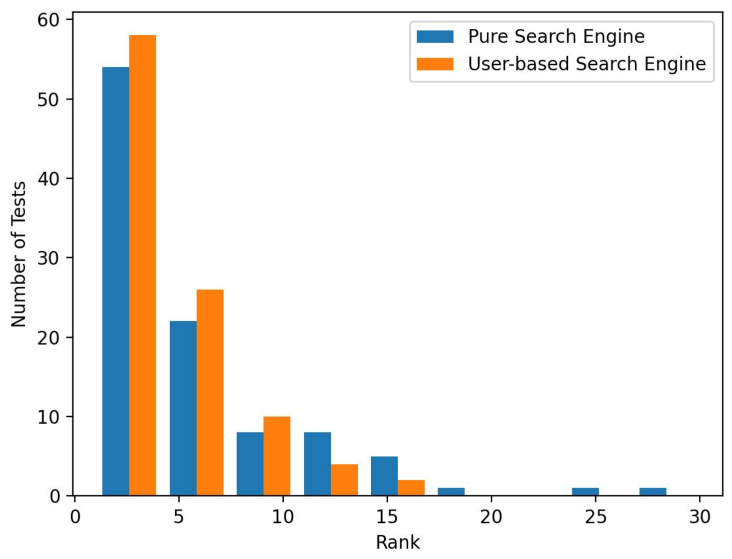

Figure 6. Ranking distribution of the corresponding systems

Conclusion

In this project, we use TF-IDF to achieve episode-specific retrieval on queries of scenes and quotes. The evaluations are performed on manually labeled test queries. A potential improvement would be aggregating relevant scripts to multiple scenes to give the specific time of target scenes.

References

1. Baptiste R. (2020). Introduction to recommender systems. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
2. https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data
3. https://www.kaggle.com/nikdavis/steam-store-games?select=steam.csv
4. Kang, W.K. & McAuley, J.M. (2018) Self-Attentive Sequential Recommendation. https://cseweb.ucsd.edu/~jmcauley/pdfs/icdm18.pdf
5. Fan S., Yang S., He S.. (2019) "Video game search engine" University of Michigan Fall 2019 Information Retrieval Project.
6. Kumar V. (2020) Single Value Decomposition (SVD) In Recommender System.