Evaluating a Hybrid Recommender System Aimed at Poetry Recommendations

I. INTRODUCTION

A. Domain of Application

The domain that this study will focus on is poetry books and poem collections. There are millions of poems, which can be categorised into various types, such as sonnets, haikus or free verse. Usually, poem anthologies are carefully curated based on a shared topic, such as a poem type, a particular theme found across poems or the author of a set of poems, and these can be used as features when implementing a recommender system.

B. Purpose/Aim

The overarching aim of this study is to produce two poetry recommender systems – one that is personalised to a user, using a hybrid approach of recommendations, and one that is non-personalised, simply recommending popular poetry books. The former will use a switching model to change between a content-based and model-based collaborative recommender system, depending on the situation. This paper will outline and evaluate the methods followed to achieve this goal to compare the performance of the two systems.

II. METHODS

A. Data Description

The data used is taken from UCSD Book Graph [1], which collated a list of books, anonymous reviews and user-book interaction data from Goodreads [2] in 2017. These are separated into three JSON formatted files [3,4].

A poetry book specific subset of this data was used, including over 35,000 books, ~2.7 million user interactions and over 150,000 reviews. This was selected due to the volume of data available and to explore a domain that many papers have not delved into. This dataset was also chosen for the use of contextual data, such as the date of when reviews were added, which can further enhance personalised recommendations.

The book data file contains a comprehensive list of details, such as a book's ISBN, the language it is in and a short synopsis of the book. The user-interaction file is a collection of explicit user ratings for books and/or whether a book had been reviewed, along with user metadata. The review data file supplements this and stores the content of reviews; these files are separated simply to reduce file size.Data Preparation and Feature Selection.

Preliminary data analysis techniques were followed from the GitHub code that came along with the data [5], to inspect the data and view random rows.

By loading the dataset with pandas, unnecessary data was cut out, leaving the columns above. Data from the "Book-Item interactions" was filtered to ensure that only reviews for books which a user had read would be considered. The "Reviews" and "Book-Item Interactions" data were merged, due to the tables sharing the same primary and foreign keys. For the use of future data processing, categorical data within columns were converted into numerical values.

Poetry Books	Reviews	Book-Item Interactions
Book ID *PK*	Book ID *FK*	Book ID *FK*
Title	User ID	User ID
Book Description	Review ID *PK*	Review ID *PK*
Authors	Rating	Rating
Genres (user categorised "shelves" in Goodreads)	Review Text	Read by the User
Image URL	Date Added	Date Added
Average user rating		

Fig 2. A table showing the three sets of data, along with their primary and foreign keys.

The complete set of data is seen in Fig 2, which also holds the item features of book description, authors, country of origin, literary genres, number of ratings and the average user rating, as well as user ratings and user reviews. These are sufficient for both collaborative filtering - as product ratings can be analysed - and also content-based filtering - due to the content within reviews, book descriptions and further potential genres.

Using the poetry books array, popular shelves were extracted from genres, in addition to the main authors of the books. Through cross-referencing a word corpus. it was ensured that each genre extracted was a valid concept. A book-features vector space model was created in preparation for content-based filtering; this was done by assigning each of these features to a new column and then marking each feature contained by each book. Additionally, a user-book array was created to mark the ratings of each book by each user, to be used for collaborative filtering.

This pre-processing was sufficient since the unnecessary data was disregarded, to focus solely on the features and variables required for the parts of the recommender system. Similarly, the data was reduced as much as possible for these algorithms.

B. Recommendation Techniques/Algorithms

A switch hybrid approach was applied, combining both content and collaborative-based filtering. To create the personalised recommendations, the content-based part analyses the extracted features of the items that a user previously engaged with – including the labels of genres and authors – in order to create a user model. With the vector space model created and having used the chi-squared method to reduce the number of features, since the chi-squared test is most common amongst categorical data [6], the values in the matrix were weighted using cosine normalisation, sufficiently fitting the values between zero and one. This prepared the prediction for unseen books, allowing cosine similarity to be easily applied. This was used since it is much more accurate than the majority of metrics [7].

The collaborative-based part allows the use of user book ratings to also create a predictive model. A model-based approach was used to find hidden shared variables. This is taken in preference to a memory-based approach, since the poetry dataset contains sparse data as there were many more items than users. With SVD, which is used since it improves with implicit feedback, a latent factor space was created; the model parameters were learned, and Stochastic gradient descent was used to improve these parameters. SVD was chosen due to simplicity, and as it can still accurately capture the relationship between the data.

Through the combination of these two approaches, many drawbacks of the individual recommender systems were avoided, such as the problem with cold starts or over-specialisation with content-based filtering - which is avoided with a model-based collaborative approach - or the necessity of item ratings for new items with collaborative-based filtering. Although a hybrid feature augmentation model works well with collaborative and content-based filtering and, as it performs the best amongst many hybrid models [8], a switching model was selected. This was chosen to overcome the issue of the sparse data found within the dataset, as poetry books are not the most popular domain (which is one reason this was chosen to be investigated as mentioned previously). As a result, new users with few ratings use the content-based recommender system, which switches to the collaborative based recommender system when there have been enough interactions and ratings. Consequently, for the approach taken in this paper is sufficient for this domain.

The personalised model is appropriate for recommending poetry, since it can consider all of the features from the reduced dataset, ensuring suitable predictions are then recommended. In comparison, the non-personalised model looks simply at the average rating of books and the most popular books amongst users to provide recommendations.

C. *Evaluation Methods*

To evaluate the performance of the two recommender systems, both the accuracy of rating predictions and the serendipity of ranking will be considered.

Predictions go on to form recommendations, and so these must be up to standard. This will be calculated through the root mean squared error (RMSE) between the actual and predicted rating values. In addition, novelty and diversity is required to create an apt recommender system, and an unexpectedness metric [9] will be calculated for both the content and collaborative parts of the personalised recommendation system. As both calculations fit well with the previously described techniques, they can effectively evaluate and compare the difference between the two recommender systems.

These evaluation metrics are sufficient when considering the domain of this application, as RMSE is very popular when evaluating linear regressions. Also unexpected metric can be used to determine whether users are shown other items they may not usually interact with, which can increase the range of type items shown to users.

III. IMPLEMENTATION

A. *Input Interface*

A command line input interface was implemented allowing the user to directly search and then rate books.

The user is instantly made aware of the data that will be collected, including the possibility of contextual data and user ratings being stored. This conforms with ethical guidelines.

For the personalised recommender system, the user is then prompted to login with their user ID. This allows user ratings to be stored and searched for when collaborative filtering is in use. If the user does not enter a valid user ID, one is generated for their use and a new user is added to the user-books matrix and the poetry table. The user is then prompted to search for a book or quit the system. Once there is a valid search, the user has the option to rate a book, which appends the user-book matrix, adjusting the collaborative part of the hybrid recommender system. To exit the system, the user can simply type ‘q’ when prompted.

For the personalised recommender system, on the other hand, the only input available is to refresh the set of the top random books and to exit the system. There is no search function for similar books and there is no user login as this is not necessary.

B. *Output Interface*

For the personalised recommender, if a book is found the user is able to view a description of the book, however if there are no similar books, the user is notified. Similarly, the user is notified if they do not enter a valid book and they are prompted to search again. Additionally, a successful search results in the top five most similar books, to the book that was searched, to be displayed.

If the user has more than 20 ratings, the hybrid recommender changes to use collaborative filtering. Similarly, if the user has interacted with the system for more than 20 iterations, the system also uses collaborative filtering. This is to help avoid the cold start problem.

For the non-personalised recommender system, a set of the most popular books were previously extracted. Five of these top books are simply displayed in a random order.

These are sufficient as they meet the requirements of this investigation by providing personalised and non-personalised recommendations, in a user-friendly manner.

IV. EVALUATION

A. *Evaluation Metrics’ Results*

The data was split into a training set and a test set (80% to 20%), to see the results of a linear regression correlation. The RMSEs were assessed on independent data points and so the following results below are in a qualitative format.

The content-based recommender system was trained on extremely sparse data and so performed extremely poorly, as it was unable to produce a sufficient number of recommendations for many users.

On the other hand, the collaborative-based recommender system was able to create quite a few recommendations and performed better on the data than the content-based recommender system.

Both of these systems struggle with serendipity and the unexpectedness of recommendations, especially the content-based recommender.

B. *Comparison*

When comparing the parts of the hybrid recommender system to the non-personalised recommender system, it is clear that the content-based recommender performs ever so slightly better. Although the cold start problem is avoided, very few recommendations are produced and so the non-personalised system would be better in this sense, since the issue of the sparsity of data is not avoided. This system could be more serendipitous in nature to provide more recommendations.

On the other hand, it is clear that the collaborative part of the recommender system is able to sufficiently adapt when it is in use, as many more recommendations were produced, which yielded a good RMSE. This is much better at providing recommendations than the non-personalised version.

V. CONCLUSION

A. Limitations

It is evident that the methods described succeed in producing poetry recommendations. However, although it was interesting to explore a sparse dataset, the data in this instance was far too sparse, making it harder to evaluate the recommender systems; a knowledge-based approach be implemented in the future to see if this creates any changes. Additionally, a more context-aware data could have been sourced, however it is likely that the volume of data collected here would have been less than the dataset that was used. Moreover, a limitation to this study as a whole is that evaluation is not completed on a wide range of models and recommender systems to further compare the system that was created.

B. Summary

In conclusion, this paper outlines the design of a personalised and non-personalised poetry book recommender system, with reasons given for each choice made. Data preparation can be carried out for the UCSD Book Graph Goodreads dataset, resulting in data for poetry books, reviews and user-item interactions. A switch hybrid model, using content-based filtering and model-based collaborative filtering, was used to create personalised recommendations, whereas popular books was used as a metric for non-personalised recommendations. Finally, to evaluate these two models, the root mean squared error and unexpectedness metrics were used. Results indicated that the collaborative filtering component of the hybrid model performed significantly better in generating accurate and diverse recommendations compared to the content-based component and the non-personalised model. However, challenges like data sparsity and limitations in serendipity were identified. Future work could explore alternative hybrid approaches or incorporate more robust contextual data to further improve recommendation quality in this niche domain.

REFERENCES

[1] "Goodreads Dataset," *UCSD Book Graph*. [Online]. Available: https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home. [Accessed: 02-Nov-2022].

[2] *Goodreads*. [Online]. Available: https://www.goodreads.com/. [Accessed: 02-Nov-2022].

[3] Mengting Wan, Julian McAuley, "Item Recommendation on Monotonic Behavior Chains", *RecSys'18*. [bibtex] [Accessed: 02-Nov-2022].

[4] Mengting Wan, Rishabh Misra, Ndapa Nakashole, Julian McAuley, "Fine-Grained Spoiler Detection from Large-Scale Review Corpora", *ACL'19*. [bibtex] [Accessed: 02-Nov-2022].

[5] Mengting Wan, "samples," *GitHub*, 29-May-2019. [Online]. Available: https://github.com/MengtingWan/goodreads/blob/master/samples.ipynb. [Accessed: 03-Nov-2022].

[6] J. Brownlee, "How to choose a feature selection method for machine learning," *Machine Learning Mastery*, 20-Aug-2020. [Online]. Available: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/. [Accessed: 05-Nov-2022].

[7] K. S. Periyasamy, K. Ponnambalam, J. Jaiganesh, and J. Rajasekar, "Analysis and performance evaluation of Cosine neighbourhood recommender ..." [Online]. Available: https://www.researchgate.net/publication/319482272_Analysis_and_performance_evaluation_of_cosine_neighbourhood_recommender_system. [Accessed: 06-Nov-2022].

[8] R. Burke, "Hybrid Web Recommender Systems." [Online]. Available: https://www.researchgate.net/publication/200121024_Hybrid_Web_Recommender_Systems. [Accessed: 06-Nov-2022].

[9] D. Kotkov, J. Veijalainen and S. Wang, "Challenges of Serendipity in Recommender Systems." [Online]. Avaliable: https://www.scitepress.org/Papers/2016/58798/58798.pdf. [Accessed: 06-Nov-2022].