LLM Based Cross Modality Retrieval to Improve Recommendation Performance

Fahad Anwaar, Adil Mehmood Khan, Muhammad Khalid, Ammad Khalil and Muhammad Awais

***Abstract*—The metadata of items and users play an important role in improving the decision-making process in the Recommender System. In recent times, web scraping-based techniques have been widely utilized to extract explicit user and item metadata from different social platforms to improve recommendation performance. Currently, Large Language Models (LLMs) have the great potential to replace the traditional web scraping-based paradigm in Recommender Systems. In this paper, we investigated the impact of LLMs and web scraping-based extraction of explicit data on the performance of the Recommender System. Firstly, a cross-modality retrieval-based LLM Gemini is explored to generate semantically enriched textual descriptions of items from digital images. The Gemini LLM is prompted with few-shot prompting on the MovieLens dataset to generate a textual description of movies based on the corresponding movie poster. Secondly, the textual descriptions for each movie in the MovieLens dataset are scraped from the OMDB API. Finally, the cross-modality retrieval-based and scraping-based textual descriptions of items are incorporated into a hybrid Recommender System to assess the quality of explicit data in terms of recommendation performance. The experimental results on the MovieLens dataset demonstrate that LLM-generated content is more effective, achieving a 0.5134 RMSE in enhancing the performance of the Recommender System.**

***Index Terms*—Recommender System, LLMs, Web Scraping, NLP, Generative Content**

I. Introduction

Recommender Systems (RecSys) is a particular type of intelligent system that utilizes consumer preferences and past user-item interaction behavior to generate personalized recommendations. The success of online platforms is immensely dependent on the performance of their recommendation algorithms in terms of user satisfaction and profit. According to [1], thirty-five percent of customers buying on Amazon and seventy-five percent of viewers on Netflix are truly influenced by their product recommendation algorithms. In general, RecSys has great potential to enhance user satisfaction, improve customer engagement, and drive business growth by providing personalized recommendations [2].

However, the performance of RecSys becomes challenging when only implicit data is available such as click-through rate, purchasing behavior, views, and frequency of interaction with specific items. To overcome this issue, explicit data is required such as item description, user reviews, tags, likes/dislikes, and user ratings [3]. In recent years, web scraping-based techniques have been widely adopted to extract explicit data from different social media platforms such as Reddit, Twitter, Facebook, OMDB, Amazon, Goodreads, and Yelp. However, web scraping-based techniques have several limitations: (a) sensitivity to data inconsistencies, (b) time-consuming process, (c) bias and fairness, (d) limited access to real-time data, and (e) legal and ethical concerns [4], [5].

In recent studies, it has been proven that textual descriptions of users and items play an instrumental role in improving the performance of RecSys. Currently, LLMs are widely used to enhance the decision-making process in Recommender Systems (RecSys) [6]. With the advent of LLMs, the content generation process in various natural language processing-based applications becomes more automatic and provides promising results. LLMS are extensively being exploited in various natural language tasks such as conversational systems, summarization, translation, and text generation. However, the LLMs are relatively less explored to assess the quality of generated textual descriptions in terms of recommendation performance.

In this paper, we aim to investigate the impact of Large Language model-generated textual descriptions and manual web crawling-based scraping of textual descriptions on the improvement of RecSys. In the first phase, we utilize the Open Movie Database (OMDB) API to manually scrape detailed textual descriptions for each movie title and year in the MovieLens dataset. In the second phase, we leverage the capabilities of cross-modality retrieval-based LLM Gemini to generate the semantically enriched textual descriptions for each movie poster in the MovieLens dataset. Finally, our primary focus will be evaluating the HRS-CE (hybrid recommender system using content embeddings [7]) with LLM-generated content and comparing it with the scraping-based sourcing method. This comparison will showcase the significance of LLM-generated descriptions in enhancing recommendation accuracy.

Additionally, the influence of LLM-generated content and scraping-based descriptions will be investigated using a range of word embedding techniques to demonstrate the effectiveness of both sources in RecSys. This research will provide a roadmap to reduce reliance on web scraping techniques and promote the utilization of LLMs for high-quality content generation through cross-modality retrieval. The LLM-generated textual descriptions not only benefit in RecSys but also various other domains.

F. Anwaar is with the Centre of Excellence for Data Science, Artificial Intelligence and Modelling (DAIM), University of Hull, Hull, HU6 7RX, U.K., A. Khan & M. Khalid are with the School of Computer Science, University of Hull, Hull, HU6 7RX, U.K., A. Khalil is with University of Education, Lahore, 54770, PK. M. Awais is with School of Computing Science, University of East Anglia, UK.

Correspondence: email: m.khalid@hull.ac.uk

The rest of the paper follows this structure: Section 2 describes the related work. The proposed methodology is discussed in Section 3. Performance evaluation is discussed in section 4. Finally, section 5 concludes this paper with insights on future work.

II. Related Work

In recent years, explicit data from several sources have been widely incorporated into recommender systems to improve user satisfaction and engagement. In [7], the hybrid framework is proposed to integrate rich content embeddings into the recommender system. The rich contextual information is scrapped from the movie database and provided as side information to predict the rating for the newly coming item into the system. In [8], a collaborative deep learning-based recommender system utilizes the explicit data of item description to address the cold start issue in the recommender system. The extracted explicit data is incorporated as a rich auxiliary description to improve the recommendation performance. In [9], the content-based model leverages the benefits of both user and item metadata to break the data sparsity and cold start ice barrier in the recommender system. The user tags and short descriptions of movie plots are utilized as explicit data in the stacked auto-encoder to improve the recommendation performance. In [10], a session-based movie recommendation system utilizes explicit data crawled from an open-source IMDbPY API to capture user behavior and patterns which helps to strengthen the recommendation system. In [11], a context-aware recommender system utilizes contextual information and user preferences to recommend the user's favorite restaurants. The user reviews are crawled from online platforms where most of the restaurants receive valuable feedback from their users. The scraped explicit user reviews are then used to extract important keywords and textual information, which helps the recommender system to learn the preferences and patterns of users about certain restaurants.

In [12], a semantic driven large-scale content-based recommendation system exploits the visual and textual modalities, particularly when an external knowledge graph is not available. The textual semantic features are scrapped from OMDB API, and visual cues are extracted from a digital image of a poster scrapped from TMDB API. The extracted visual semantics are integrated with textual features to formalize a multimodal representation of the item, and finally cosine similarity measure is used to determine the relevance of items that are most similar to the target user preferences. In [13], a Bayesian Sentiment analysis-based Recommender System (BayesSentiRS) is proposed to address the cold start and data sparsity issue in a ranking-based recommender system. Most of the existing work lacks visual and sentiment information aids in conventional Bayesian Personalized Ranking (BPR) approaches, relying solely on implicit feedback. BayesSentiRS introduced a novel multimodal technique to integrate both visual and semantic sentiments into BPR, enhancing the extraction of nuanced user preferences and improving the ranking-based recommendation process. In [14], a web crawler is utilized to extract the abstracts and essential information from the home pages of chosen journals or conferences, and this scrapped explicit information serves as training data for the recommender system. The training data is continuously updated by extracting details from manuscripts to guarantee precise recommendations for authors.

In [15], LLMs are explored to perform conversational recommendation tasks in zero-shot settings. The explicit data of user conversations is scrapped from a popular discussion forum Reddit. The crawled explicit data from this real-world platform provides a valuable source to investigate the effectiveness of LLMs. The extracted explicit data provides more realistic and diverse interactions to evaluate the performance of LLM-based conversational recommender systems. In [16], the Alpaca-LoRa Large Language Model is used to generate textual descriptions of books as auxiliary information for the recommendation system. However, the Alpaca-LoRa LLM is vulnerable to hallucination issues while generating the textual descriptions of books, which results in a lack of recommendation performance.

Overall, in the literature, explicit data is usually obtained from the web crawling method and the scrapped explicit data is incorporated as side information into a recommender system. To the best of our knowledge, cross-modality retrieval-based explicit data from LLM has not been studied in a recommender system, which may provide a more promising result in comparison with traditional scraping-based techniques.

III. Methodology

A cross-modality retrieval-based and scraping-based explicit data is studied in the recommender system to gauge the impact of both approaches. In this work, we have used our hybrid recommender system HRS-CE [7] method to claim the impact of cross-modality retrieval-based content generation from the LLM. The performance of the recommender system will showcase the significance of both LLM-based and scraping-based explicit data extraction techniques in the settings of cold start item scenarios. We have categorized our work into two components: (a) Traditional scraping-based explicit data in RecSys and (b) LLM-based explicit data in RecSys. The overall system framework is demonstrated in Fig 1 where the left side represents the scraping-based approach and the right side represents the LLM-based cross-modality retrieval technique.

To demonstrate the effectiveness of scraping-based and LLM-based explicit data in RecSys, the experimental results are performed on benchmark real-world MovieLens dataset. In the scraping-based component, the movie title and year from the MovieLens dataset will be used to crawl the textual description of each item from the open movie database (OMDB) API. In LLM based component, the digital image of each item in the MovieLens [17] dataset will be used to generate the textual descriptions. Then, the textual descriptions obtained from both sources will be exploited in HRS-CE [7], to assess the quality and coherence of cross-modality retrieval-based content and their potential for automating the explicit information extraction process in RecSys.

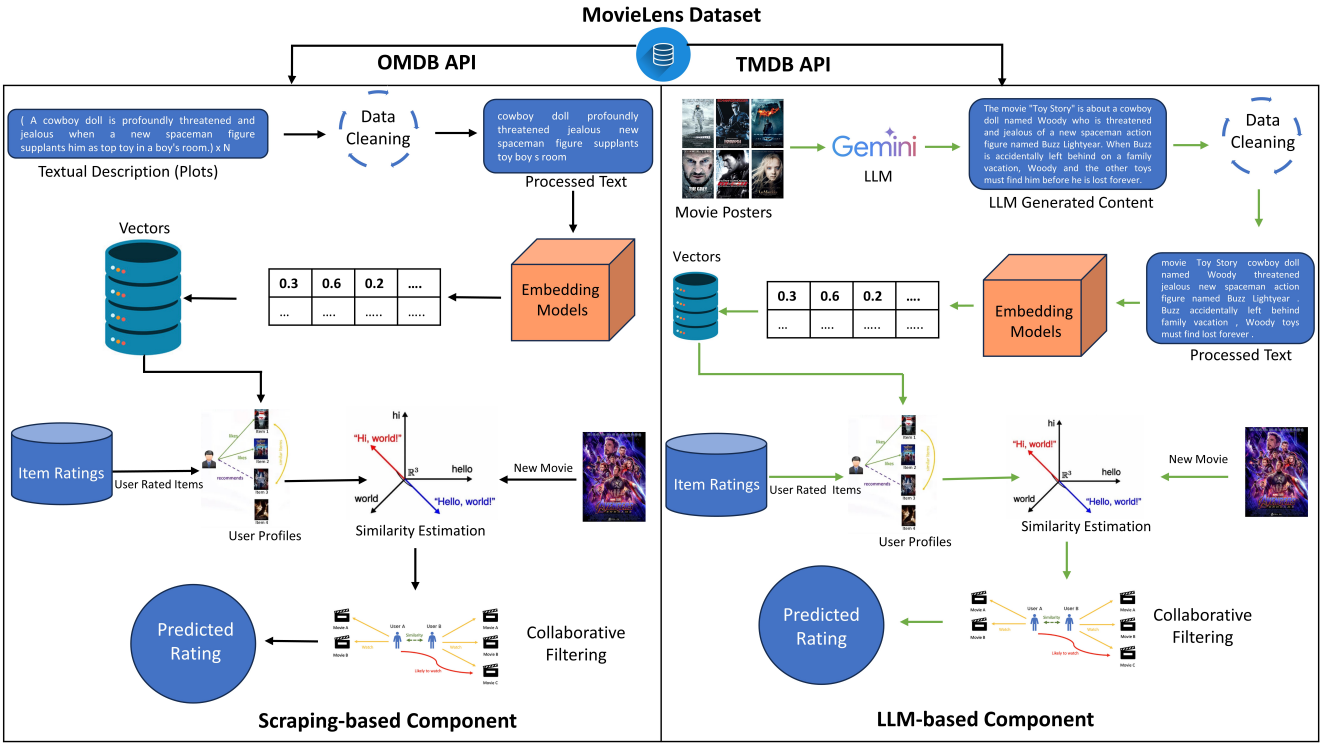

Fig. 1: Proposed System: HRS-CE (LLM)

A. LLM based Cross Modality Content Generation

In the LLM-based component, digital images of candidate items are obtained from TMDB API [1]. The movie name is concatenated with its corresponding digital poster and serves as the prompt for the LLM. The Gemini Pro Vision LLM [18] (developed by Google) is used due to its ability to handle complex multimodal data including text, image, audio, and video within a unified embedding space. The Gemini LLM offers the longest context window length that enables the processing of larger chunks of data simultaneously and employing efficient attention mechanisms which provides deeper insights into the understanding of complex tasks. The Gemini Pro Vision [18] LLM is prompted with few-shot prompting on the MovieLens dataset to generate a textual description of movies based on corresponding movie posters. The generated textual descriptions are processed using pre-processing techniques such as stop word removal (e.g., "You", "The"), which usually do not provide any useful information. The processed textual descriptions are used to produce vector representation by using different embedding models including Word2vec, GloVe, and FastText. The resultant feature vectors of items and associated user ratings are incorporated to compute the user profiles for each user in the dataset. The computed user profiles efficiently capture and depict the users' interests and preferences, providing comprehensive insights into their behaviors and likings. Then, the cosine similarity measure is used to compute the top neighborhood for cold-start items from existing user profiles in the database. The cold-start problem refers to generating recommendations for newly introduced items without previous interaction history. Finally, a memory-based collaborative filtering technique is applied to the top-selected neighbors to predict the system's rating for a cold-start item. In this work, the LLM-based contextual descriptions are exploited as auxiliary item descriptions in our HRS:CE [7] framework (same as in section B) to improve the recommendation accuracy for cold-start items in the system.

B. Scraping-based Explicit Data Extraction

In the Scrap-based component, a Python script is written to crawl the textual description of movies from OMDB API [2]. The textual descriptions are obtained for each item in the dataset, providing a valuable source for semantic-driven recommendation. The scrapped textual descriptions are utilized in the same way as we exploited in our previous work [7] to generate personalized recommendations.

C. Dataset Preparation and Experimental Setup

The performance of the proposed framework is evaluated on MovieLens 100k and 20M dataset which is considered a more stable and benchmark dataset to evaluate the Recommender System. The MovieLens dataset has been used around 22% [19] in the experiments of Recommender Systems, which shows the widespread recognition of the chosen dataset. The MovieLens 100k dataset includes 1682 movies and one hundred thousand explicit ratings on a scale of 1 to 5 provided by 943 users. The MovieLens 20M dataset comprises 27,278

[1]https://www.themoviedb.org/

[2]https://www.omdbapi.com

movies and 20 million ratings on a scale of 0.5 to 5 contributed by 138,493 users along with 465,564 tags. The original MovieLens dataset lacks textual descriptions for the movies which can be beneficial to enhance the semantic understanding of each item within the dataset. Therefore, the primary dataset is extended by involving the short textual description of movies. The short textual descriptions of movies are obtained by using scraping-based and LLM-based technique.

In the scraping-based technique, a short textual description of each movie is obtained by traversing the movie title and year in the dataset and subsequent textual descriptions of movies are scrapped from OMDB API by sending the search request. In this web crawling, we successfully obtain the textual descriptions for 96.51% movies in the dataset. Additionally, we have used the TMDB API to obtain the digital images of movies with sufficient resolution and it successfully provides 98.35% digital images for movies in the dataset. Those movies are simply discarded whose digital image or textual description is not available. The digital image of each movie in the dataset is exploited in LLM to generate the textual descriptions.

The items in the dataset are randomly shuffled before dividing into the training and test sets. We used the 80-20 rule to split the items for training and testing purposes. The representative users are determined based on the training set, and the rating of a newly coming item for exiting users (cold-start item) is predicted for test set items.

IV. Performance Evaluation

This section describes the performance of both explicit data sources in the Recommender System. The evaluation metrics are discussed, followed by results and discussion with other techniques.

A. Evaluation Metric

The performance of the proposed system in both scraping-based and LLM-based components is measured using the Root Mean Square Error (RMSE) metric.

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,x}(R_{ux} - \hat{R}_{ux})^2} \quad (1)$$

where N indicates the total number of predicted ratings for the newly coming item (cold start) in the system, $\hat{R}_{ux}$ represents predicted rating, and $R_{u\hat{x}}$ indicates known rating on particular item i specified by user u. The RMSE score near zero means greater accuracy in recommendations.

B. Results and Discussion

The performance of both explicit data sources in RecSys is assessed through 5-fold cross-validation using MovieLens 20M and 100k datasets. The experiments are performed to assess the impact of K number of nearest neighbors. The value of K ranges from 0 to 100 which represents the most identical users in terms of rating prediction task. The score of RMSE significantly minimizes as the number of nearest neighbors increases. The consideration of a greater number of similar neighbor users for predictions helps to improve the overall recommendation accuracy, as evident from Fig 2 and Fig 3.

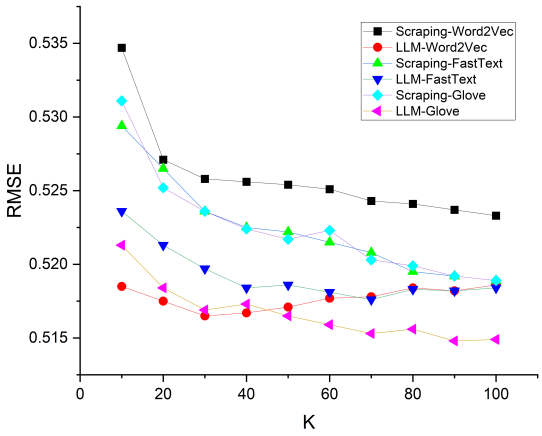

Fig. 2: Performance on 20M dataset with varying neighbors

The experimental results are obtained using a range of embedding methods to investigate the impact of explicit data sources. The performance comparison of both explicit data sources in RecSys using the MovieLens 20M dataset is shown in Table 1. The scraping-based explicit features in RecSys obtained the highest RMSE of 0.5347 at K=10 and achieved the optimal result of 0.5233 RMSE at K=100 by using word2vec. The LLM-generated textual descriptions reported the highest RMSE of 0.5186 at k=100 and performed better at k=30 by achieving a 0.5165 RMSE with word2vec. With FastText embeddings, the scraping technique reported the highest RMSE of 0.5294 at K=10 and performed better with the highest number of neighbors at k=100 with 0.5189 RMSE. Meanwhile, the LLM-based technique obtained a higher RMSE of 0.5236 at K=10 and reported optimal results at k=70 with 0.5176 RMSE. The scraping method with Glove embeddings obtained the optimal result of 0.5189 RMSE at K=100, while the LLM-based approach achieved the ideal performance on k=90 with 0.5148 RMSE.

The performance of scraping and LLM-based techniques is also evaluated using the MovieLens 100k dataset, as shown in Table 2. In the scraping-based technique, the optimal result of 0.5230 RMSE was reported at K=100, while the LLM-based technique achieved better performance at K=50 with a 0.5152 RMSE using word2vec. With FastText embeddings, the scraping-based method obtained the optimal RMSE of 0.5219 at k=100, while the LLM-based approach reported even better results at k=80 with an RMSE of 0.5158. The scraping method with Glove embeddings achieved the optimal result of 0.5190 RMSE at k=80, while LLM reported the ideal performance on K=60 with 0.5134 RMSE. In the collaborative filtering component of HRS:CE [7], K represents the number of nearest neighbors/users with similar tastes or preferences, which is particularly important in sparse datasets. Increasing the value of K indicates incorporating more user preferences, thereby enhancing prediction accuracy in HRS-CE [7]. A

TABLE I: Performance Comparison of Explicit Data in RecSys using MovieLens 20M Dataset

HRS-CE	Embedding	RMSE									
		k=10	k=20	k=30	k=40	k=50	k=60	k=70	k=80	k=90	k=100
Scraping	Word2Vec	0.5347	0.5271	0.5258	0.5256	0.5254	0.5251	0.5243	0.5241	0.5237	**0.5233**
LLM	Word2Vec	0.5185	0.5175	**0.5165**	0.5167	0.5171	0.5177	0.5178	0.5184	0.5182	0.5186
Scraping	FastText	0.5294	0.5265	0.5236	0.5225	0.5222	0.5215	0.5208	0.5195	0.5192	**0.5189**
LLM	FastText	0.5236	0.5213	0.5197	0.5184	0.5186	0.5181	**0.5176**	0.5183	0.5182	0.5184
Scraping	Glove	0.5311	0.5252	0.5236	0.5224	0.5217	0.5223	0.5203	0.5199	0.5192	**0.5189**
LLM	Glove	0.5213	0.5184	0.5169	0.5173	0.5165	0.5159	0.5153	0.5156	**0.5148**	0.5149

TABLE II: Performance Comparison of Explicit Data in RecSys using MovieLens 100k Dataset

HRS-CE	Embedding	RMSE									
		K=10	K=20	K=30	K=40	K=50	K=60	K=70	K=80	K=90	K=100
Scraping	Word2Vec	0.5335	0.5270	0.5258	0.5256	0.5254	0.5250	0.5243	0.5240	0.5237	**0.5230**
LLM	Word2Vec	0.5207	0.5172	0.5163	0.5159	**0.5152**	0.5152	0.5154	0.5153	0.5154	0.5152
Scraping	FastText	0.5371	0.5303	0.5275	0.5261	0.5248	0.5237	0.5232	0.5227	0.5223	**0.5219**
LLM	FastText	0.5163	0.5161	0.5164	0.5175	0.5159	0.5178	0.5169	**0.5158**	0.5161	0.5164
Scraping	Glove	0.5293	0.5238	0.5224	0.5210	0.5205	0.5209	0.5191	**0.5190**	0.5198	0.5193
LLM	Glove	0.5189	0.5182	0.5161	0.5164	0.5135	**0.5134**	0.5137	0.5138	0.5137	0.5139

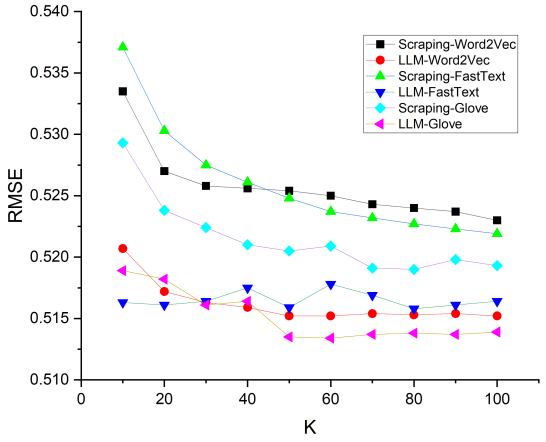

Fig. 3: Performance on 100k dataset with varying neighbours

smaller value of K cannot capture sufficient context, resulting in higher RMSE (low accuracy). A larger K value includes more contextual depth and improves prediction performance with lower RMSE (high accuracy). However, there is an optimal value for K (underline and bold in TABLE I and TABLE II); exceeding this threshold leads to performance degradation due to the inclusion of irrelevant or noisy users. The Word2Vec-based embeddings improve the results with larger K values as they only capture the local context well. However, they do not incorporate sub-word information and struggle with rare words in the corpus. In contrast, FastText embeddings incorporate the sub-word information but perform poorly at small K values because they only focus on the sub-words and neglect the broader contextual information. The Glove embeddings perform best across varying numbers of K neighbors due to their ability to capture local and global contexts more efficiently.

The performance of proposed LLM-based explicit data sources is also compared with other techniques in literature to predict the rating for newly coming items in the Recommender System. Table 3 summarizes the performance of the proposed method with other best-performing models for cold start item recommendations.

TABLE III: Comparison with other models.

Model	Dataset	RMSE
Koren et al. [20]	20M	0.8721
Nguyen et al. [21]	20M	0.8528
Strub et al. [22]	20M	0.7652
Anwaar et al. [7]	100k	0.5217
Anwaar et al. [7]	20M	0.5272
HRS-CE (LLM)	100k	**0.5134**
HRS-CE (LLM)	20M	**0.5148**

Koren et al. [20] proposed an optimized SVD model (SVD++), which incorporates both implicit and explicit responses in RecSys to forecast ratings for unknown items and obtained a score of 0.8721 RMSE on the dataset of MovieLens 20M. Nguyen et al. [21] deliver the unified model that combines the implicit and explicit user feedback to predict the rating and achieved a score of 0.8528 RMSE for the 20M dataset. Strub et al. [22] proposed a V-CFN++ model based on a collaborative neural network approach, which utilizes the sparse input ratings and side information to compute the non-linear matrix factorization and obtained a score of 0.7652 RMSE using 20M dataset. Anwaar et al. [7] proposed HRS-CE which incorporates the content embeddings in RecSys to predict the rating for newly coming or unknown items. The HRS-CE model significantly outperforms the other models in literature by introducing the semantically enriched textual description which helps to outperform the other explicit and implicit feedback-based techniques in the Recommender System. The HRS-CE model achieved a score of 0.5217 and 0.5272 RMSE on the 100k and 20M datasets respectively. However, our proposed LLM-based explicit data source approach in RecSys obtained even better results than the traditional scraping-based technique and achieved an accuracy of 0.5134 and 0.5148 RMSE on the 100k and 20M datasets respectively.

The cross-modality retrieval-based content generation from LLM is comparatively better than the traditional scraping approach in RecSys. However, it is evident from the results that scrapped-based data sources are more structured which helps to continuously improve the recommendation accuracy

with varying numbers of k similar neighbors. On the other hand, the recommendation accuracy is more fluctuated with LLM-based data sources as the number of neighbors increased. In the proposed system, the textual description obtained from the scraping method is more concise, while the LLM-based method provides a more detailed textual description with a wider scope. The LLM-generated detailed textual descriptions capture the nuanced aspects of new movies in the system, better aligning with user preferences and providing comprehensive item representation. Consequently, detailed textual descriptions obtained from LLM facilitate the formulation of the enriched semantics of candidate items, resulting in improved user-profiles and potentially leading to more accurate recommendations.

V. Conclusion

In this paper, a cross modality retrieval based, and web crawled explicit information is studied in Recommender System (RecSys). The textual descriptions obtained from both sources are incorporated in hybrid RecSys to forecast the rating for a newly coming item in the system also known as the cold start item problem. We investigate two techniques in our research: (a) scraping textual descriptions from the OMDB API by sending the search request for each item, and (b) using a cross-modality retrieval-based LLM called Gemini Pro Vision to generate a textual description of movies based on the corresponding digital image. The performance of both explicit sources is investigated in HRS-CE using MovieLens 100k and 20M datasets. The experimental results indicate that LLM-generated textual descriptions have competitive results and can be used in place of web scraping methods. The obtained results (0.5134 and 0.5148 RMSE) highlight the potential of cross-modality retrieval-based LLM to automate the content generation process in RecSys and minimize the reliance on the manual scraping method. The LLM-generated content also removes personal biases of scrapped websites and thus provides a new roadmap to content providers to save time and effort. However, the content analysis and performance metric reveal that LLM-generated content exhibits more variability in results when adjusting the number of k nearest neighbors. In contrast, scraping-based content performance consistently improves with an increased number of k nearest neighbors. Consequently, the content generated by LLM is less structured and contains a broader scope of candidate items, resulting in more fluctuations in recommendation accuracy with varying numbers of neighbors.

In the future, we aim to study the impact of LLM-generated content in other domains such as books, TV shows, research articles, newspapers, and job recommendations. Additionally, we will explore other multimodal LLMs, such as open-source options like LLaMA-VID and InternVL 1.5, to evaluate their multimodal content generation capabilities and their effect on improving recommendation accuracy. We will also focus on the impact of other cross modalities such as audio-to-text, video-to-text, and text-to-image-based explicit content generation from LLM and their potential to address the data sparsity and system cold-start problem in the Recommender System.

References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*. Springer, 2010, pp. 1–35.

[2] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.

[3] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert systems with applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[4] N. Kumar, M. Gupta, D. Sharma, and I. Ofori, "Technical job recommendation system using apis and web crawling," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[5] V. Krotov and L. Silva, "Legality and ethics of web scraping," 2018.

[6] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, X. Li, C. Zhu, H. Guo, Y. Yu, R. Tang *et al.*, "How can recommender systems benefit from large language models: A survey," *arXiv preprint arXiv:2306.05817*, 2023.

[7] F. Anwaar, N. Iltaf, H. Afzal, and R. Nawaz, "Hrs-ce: A hybrid framework to integrate content embeddings in recommender systems for cold start items," *Journal of computational science*, vol. 29, pp. 9–18, 2018.

[8] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.

[9] F. Anwar, N. Iltaf, H. Afzal, and H. Abbas, "A deep learning framework to predict rating for cold start item using item metadata," in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, 2019, pp. 313–319.

[10] M. Potter, H. Liu, Y. Lala, C. Loanzon, and Y. Sun, "Gru4recbe: A hybrid session-based movie recommendation system (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 13 029–13 030.

[11] V. Boppana and P. Sandhya, "Web crawling based context aware recommender system using optimized deep recurrent neural network," *Journal of Big Data*, vol. 8, pp. 1–24, 2021.

[12] M. M. Bendouch, F. Frasincar, and T. Robal, "A visual-semantic approach for building content-based recommender systems," *Information Systems*, p. 102243, 2023.

[13] L.-H. Wu, "Bayessentirs: Bayesian sentiment analysis for addressing cold start and sparsity in ranking-based recommender systems," *Expert Systems with Applications*, vol. 238, p. 121930, 2024.

[14] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowledge-Based Systems*, vol. 157, pp. 1–9, 2018.

[15] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, and J. McAuley, "Large language models as zero-shot conversational recommenders," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 720–730.

[16] A. Acharya, B. Singh, and N. Onoe, "Llm based generation of item-description for recommendation system," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1204–1207.

[17] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[18] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[19] M. M. Khan, R. Ibrahim, and I. Ghani, "Cross domain recommender systems: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–34, 2017.

[20] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[21] T. Nguyen and A. Takasu, "A probabilistic model for the cold-start problem in rating prediction using click data," in *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part V 24*. Springer, 2017, pp. 196–205.

[22] F. Strub, J. Mary, and R. Gaudel, "Hybrid collaborative filtering with autoencoders," *arXiv preprint arXiv:1603.00806*, 2016.