Deep Learning-Based Recommender System for Arabic Content with Integrated Sentiment Analysis of User Reviews

Amani Al-Ajlan[1], Nada Alshareef[2]
Department of Computer Science-College of Computer and Information Sciences,
King Saud University, Riyadh, Saudi Arabia[1]
Department of Information Technology-College of Computer and Information Sciences,
King Saud University, Riyadh, Saudi Arabia[2]

***Abstract*—Recommender systems are widely used as an information filtering technology to automatically predict and identify a set of interesting items for users based on their needs and preferences. They are widely applied in many domains, including e-commerce, social media, education, and healthcare. Recommender systems employ various filtering approaches, such as collaborative filtering, content-based filtering, and hybrid methods. Collaborative filtering is broadly categorized into memory-based and model-based approaches. Deep learning-based recommenders are a type of model-based approach that employs neural networks to capture patterns in user preferences and item features and generate accurate and personalized recommendations. In this study, we apply deep learning-based recommender systems to the Large-Scale Arabic Book Reviews Dataset (LABR) and evaluate their performance. To improve recommendation quality, we integrate sentiment analysis of user reviews using pre-trained Arabic BERT–mini and AraBERT, enabling more accurate modeling of user preferences. The results show that the combination of deep learning techniques and sentiment analysis produces more accurate recommendations, improving user satisfaction and engagement with Arabic content.**

Keywords*—Recommender system; deep learning; sentiment analysis; LABR dataset; Model-based collaborative filtering; pre-trained model Arabic BERT; pre-trained model AraBERT

I. INTRODUCTION

Recommender systems are effectively used as an information filtering technique to automatically predict and identify a set of interesting items for users according to their needs and preferences. Recommender systems are widely used across various applications due to the significant benefits they offer to both users and companies. They reduce the effort required to search for relevant items, thereby saving users time. Today, recommendation systems are essential parts of most applications such as Netflix, YouTube, and TripAdvisor. Recommender systems mechanisms have been shown to enhance the efficiency and consistency of decision-making. Given the vast amount of available information, these systems address the problem of information overload by filtering the user options and recommending the most suitable items, thereby improving user experience [1], [2]. There are various filtering approaches used in recommender systems, such as content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering (HF). Content-based filtering uses domain knowledge to recommend items similar to those a user has previously liked. Using user profiles and item metadata, such as item descriptions, content-based recommender systems generate personalized recommendations. This approach can be implemented either explicitly, based on user-stated preferences, or implicitly, by learning from user interactions and behavior, such as clicking on links [2]. Collaborative filtering generates recommendations based on the preferences of similar users, often referred to as “neighbors” [3]. Neighbor-based algorithms are typically used in collaborative recommender systems that model relationships either between users or between items. However, collaborative recommenders suffer from the cold-start problem, which occurs when there is insufficient data for a new user or item [4]. Various collaborative filtering methods have been proposed to generate user-desired recommendations, with memory-based and model-based approaches being the most widely used [5]. Hybrid filtering is the combination of collaborative filtering and content-based filtering to obtain the best recommendation results [6]. Recommender systems are based on methods such as nearest neighbor and matrix factorization. However, deep learning has been successfully applied in various domains, including image recognition, computer vision, speech recognition, and natural language processing, and has recently been adopted in recommender system research. To improve the quality of recommendations, many businesses have adopted deep learning-based recommender systems [1]. These systems provide a significant improvement over traditional models by effectively handling high-dimensional data and addressing complex problems [1], [7]. Moreover, deep learning models are capable of capturing non-linear relationships between users and items [6].

In this study, we propose a deep learning-based collaborative filtering algorithm, which is a type of model-based collaborative filtering algorithm that utilizes neural networks. Unlike traditional methods such as matrix factorization, deep learning techniques can model non-linear relationships, sequential behaviors, and additional data sources, including product descriptions, user reviews, and images. Consequently, the proposed system can generate more personalized recommendations.

Recently, sentiment analysis of user reviews has been integrated with recommender systems to enhance the recommendation quality and address the data-sparsity problem in traditional recommender systems [8]. In such approaches, both user ratings and textual reviews are used to improve recommendations. Sentiment analysis extracts the emotional tone of user reviews to determine user preferences. By combining sentiment analysis with traditional methods such as content-based and collaborative filtering, hybrid recommender systems can be developed that provide more relevant and personalized recommendations [9]. The systems can produce more relevant recommendations by understanding user opinions, which increases user satisfaction and engagement [10]. The cold start issue, which arises when there is insufficient data for the new item or user, can also be resolved via sentiment analysis. Even without explicit ratings, sentiment analysis can identify preferences by analyzing text data. In our study, we propose a hybrid approach that combines sentiment analysis of Arabic user reviews with deep learning-based recommender systems to improve recommendation quality and solve the cold start problem. We use the Large-Scale Arabic Book Reviews Dataset (LABR) to evaluate the performance of the proposed approach. Our system takes book reviews as input and provides the corresponding sentiments for each of the books in our dataset. Then, both user ratings from LABR and sentiment scores are used to produce more personalized recommendations. This study extends our previous work [11], in which we proposed integrating sentiment analysis of user reviews with recommender systems for Arabic content. In that work, we developed two collaborative filtering-based recommender systems: the first is based on user ratings, while the second enhances recommendations by integrating sentiment analysis from user reviews. In the sentiment analysis phase, we compared three approaches: the Arabic BERT-mini model, the AraBERT model, and the Mazajak tool. In our previous work, we employed traditional collaborative filtering methods, including Singular Value Decomposition (SVD), K-Nearest Neighbors (KNN), and Non-Negative Matrix Factorization (NMF). This study builds on that work by employing deep learning–based recommender systems. The main contributions of our research are the following: 1) Proposed a hybrid approach that combines a deep learning-based collaborative recommender system with different Arabic sentiment analysis models and compares their results. 2) Applied the Arabic BERT–mini model and AraBERT sentiment analysis model to user reviews of the LABR dataset. 3) Studied the impact of using sentiment analysis models with a deep learning-based collaborative recommender algorithm on recommendation accuracy. The remainder of this study is organized as follows: Section II reviews related work, Section III presents the methodology, Section IV discusses the results, and Section V concludes the study with future work.

II. Related Work

A. Arabic Recommender Systems

We reviewed the literature and identified several studies that investigated recommender systems for Arabic content. First, Al-Ajlan and AlShareef [12] reviewed existing studies on recommender systems for Arabic content, summarizing them based on type, domain, datasets, and sentiment analysis integration. They noted that the most recommender systems for Arabic content use collaborative filtering, and some integrate sentiment analysis, which leads to higher accuracy. The most commonly used algorithm in these studies is Singular Value Decomposition (SVD). They also reported that the largest dataset used in these studies is LABR, which contains over 63,000 book reviews. Another study by Sallam et al. [13] proposed two approaches: memory-based collaborative filtering and model-based collaborative filtering using Singular Value Decomposition (SVD). They used the LABR dataset to evaluate their proposed approach. The proposed approach improved the quality of the recommendations for Arabic content and achieved low errors in terms of RMSE and MAE. Al-Ajlan and AlShareef [11] proposed integrating sentiment analysis of user reviews with traditional collaborative filtering recommender systems to improve the quality and accuracy of recommendations. They employed memory-based, including user-user and item-item models, and model-based collaborative filtering techniques, including SVD, KNN, and NMF. The LABR dataset was used to evaluate the proposed approach. The results indicate that the recommendation quality improves when recommender systems are integrated with user review sentiment analysis. The recommender system based on SVD, combined with the Arabic BERT-mini model, achieved the best performance in terms of RMSE and MAE. Additionally, Harrag et al. [14] explored the integration of sentiment analysis with recommendation systems by predicting product ratings from users' reviews using NLP and data mining techniques. In the sentiment analysis phase, they used an SVM model with the term frequency–inverse document frequency (TF–IDF) matrix. In the recommendation phase, they applied model-based collaborative filtering using SVD. They used the Opinion Corpus Arabic (OCA) dataset to evaluate the model. The results showed that the accuracy of the system is about 85% in predicting the ratings. Another study by Srifi et al. [15] evaluated the performance of some modern recommender systems when applied to Arabic content. They applied five modern recommendation systems that were initially intended for English content (ALFM, A3NCF, PARL, CARL, and CARP) and tested them on four Amazon English datasets translated into Arabic using Google Translate. Results showed that the systems performed well on Arabic content, with accuracy comparable to their performance on English data.

B. Deep Learning-Based Recommender Systems

This section reviews prior work on deep learning-based collaborative recommender systems across various domains. Li et al. [16] proposed the first system for collaborative filtering (CF) that integrates deep learning models. They introduced a deep architecture for CF that combines deep feature learning with matrix factorization. While matrix factorization-based CF methods can effectively capture the implicit relationships between users and items, they still face challenges such as the cold-start problem and data sparsity. Li et al. [16] employed neural network algorithms such as the stacked denoising autoencoder (SDAE), which enables the learning of high-level representations. However, SDAE has several disadvantages, one of which is the high computational cost of training. To address this issue, they used a variant of SDAE known as the marginalized denoising autoencoder (mDA), which avoids high computational expense by marginalizing out the random feature corruption and provides a closed-form solution for learning model parameters. They selected three challenging tasks—movie recommendation, book recommendation, and response prediction—to evaluate the effectiveness of their proposed mDA-CF and mSDA-CF approaches. Their experiments were conducted on four datasets: MovieLens-100K, MovieLens-1M, Book-Crossing, and Advertising. The results demonstrated that the mDA-CF and mSDA-CF approaches outperformed comparable methods in all three tasks.

To facilitate book discovery on library websites with a large collection of books, Liu and Gao [17] proposed a new recommendation algorithm that integrates deep learning with matrix factorization. Their model employs a neural network to process the user-book rating matrix, aiming to address the sparsity problem commonly found in such datasets. The rating matrix is fed into a deep matrix factorization framework to learn the hidden features of users and books, which are then used to predict user ratings. Liu and Gao [17] trained and evaluated their model on the Goodbooks-10k dataset. The results show that the proposed algorithm effectively mitigates data sparsity and performs well on real-world book datasets.

Lin and Chi [18] proposed a movie recommendation system based on collaborative filtering and neural networks. In their study, a neural network was trained to predict user ratings for movies. They used the MovieLens ml-100k dataset, which contains ratings from 943 users on 1,682 movies, along with basic user and movie information. The input features included user ID, gender, age, occupation, movie ID, and movie category. The neural network architecture consisted of two hidden layers: the first with 192 neurons using ReLU as the activation function, and the second with 300 neurons using TanH. The final predicted rating was generated by the output layer. In the evaluation, the neural network-based model achieved a Mean Absolute Error (MAE) of 0.76, with a processing time of 136 seconds.

Healthcare recommender systems provide users with mobile healthcare services that are continuously accessible. However, designing patient-centered healthcare recommender systems presents several challenges, including processing the large volume of data generated by smart devices and sensors, managing dynamic networks for real-time data transmission, and the lack of effective methods for knowledge aggregation. To address these challenges, Aujla et al. [19] proposed DLRS, a deep learning-based recommender system that leverages Software-Defined Networking (SDN) within a smart healthcare environment. For evaluation, the system was tested using a dataset that included 100 patients, 100 sensors, and 20 doctors. DLRS operates through several key steps: first, a tensor-based dimensionality reduction algorithm is applied to remove irrelevant features from the collected data; second, a decision tree-based classification method groups patient queries according to different diseases; and finally, a convolutional neural network is used to generate personalized health recommendations. The system was evaluated through a case study using multiple performance metrics, including dimensionality reduction ratio, approximation ratio, accuracy, RMSE, MAPE, and latency. The results show that DLRS outperforms existing systems in its category.

III. Material and Method

A. Dataset

We used the Large-Scale Arabic Book Reviews (LABR) dataset, one of the largest sentiment analysis datasets for the Arabic language. It consists of over 63,257 book reviews, each rated on a scale of 1 to 5 stars [20]. These reviews were submitted by 16,486 users for 2,131 different books [20]. Table I summarizes the main characteristics of the LABR dataset. Each row in the LABR dataset represents a single book review submitted by a Goodreads user. Every row includes a user's rating on a scale of one to five, the Goodreads.com review id, the Goodreads.com user id, the Goodreads.com book id, and the text of the review as presented in Table II.

TABLE I. LABR Dataset Statistics

Number of reviews	63,257
Number of users	16,486
Avg. reviews per user	3.84
Median reviews per user	2
Number of books	2,131
Avg. reviews per book	29.68
Median reviews per book	6

TABLE II. Samples from LABR Dataset

Rating	Review id	User id	Book id	Review
5	279844173	6295076	2939264	كتاب اكثر من رائع ويستحق القراءة اكتر من مرة
5	322587552	8509913	2939264	قيم جداً . كل قترة اقرء احد ابوابه لانه هذا الكتاب لايتطلب قراءة سريعة بل دراســـــة جيدة
4	510316957	3665588	2939264	لا يمر عليك يوم بدون تصفحه. أعتبره زاد المسلم اليومي

B. Methodology

In this section, we will present our approach that combines a deep learning-based collaborative filtering algorithm with sentiment analysis of users' reviews. Our study builds on our earlier work [11] that integrated sentiment analysis of user reviews into traditional collaborative filtering. The proposed approach consists of four main phases: data preprocessing, sentiment analysis of users' reviews, combining the sentiment score with the user rating score, and applying deep learning-based collaborative recommender systems, as presented in Fig. 1.

1) Preprocessing and cleaning: The dataset was preprocessed and cleaned using "Re" and "Pandas" Python libraries [21], [22]. The preprocessing step included normalizing Arabic characters and removing repeated letters, punctuation marks, and diacritics from the text. Then, the dataset was divided into positive, negative, and neutral categories based on the rating. Books with 4 or 5 stars were considered positive, those with 3 stars were neutral, and those with 1 or 2 stars were categorized as negative.

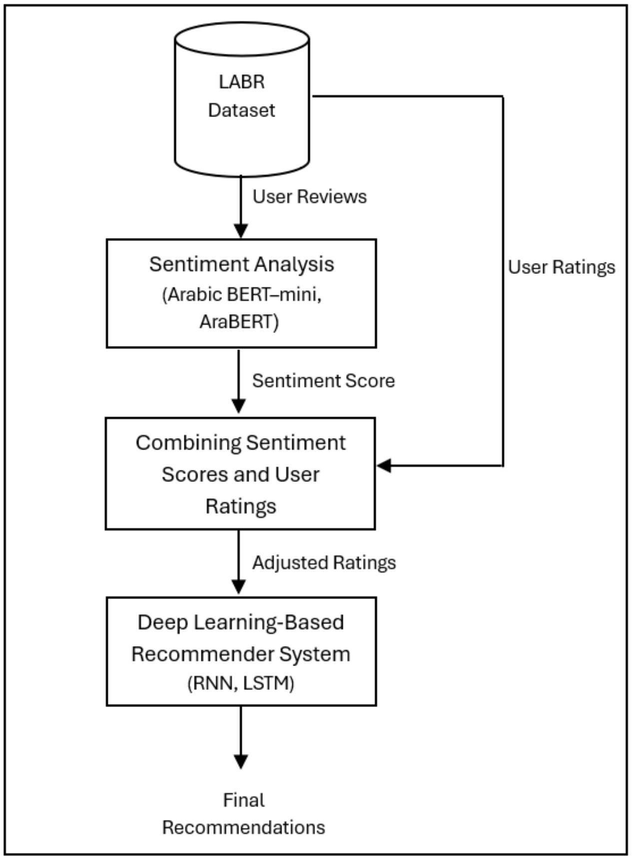

Fig. 1. The proposed approach.

2) Sentiment analysis: we applied two sentiment-analysis approaches using pre-trained Arabic language models: Arabic BERT–mini model and AraBERT [23], [24], [25]. The Arabic BERT–mini model [24], [26] was originally trained on a large Arabic corpus of about 8 billion words from OSCAR [27], Arabic Wikipedia [28], and other large text sources (around 95 GB). We fine-tuned this model using the Arabic Sentiment Twitter Corpus [29] which contains 58K Arabic labeled tweets. We also experimented with several additional datasets—SS2030 [30], the 100K Reviews dataset [31], and ArSAS [32]—and found that the Arabic Sentiment Twitter dataset produced the best performance. After tuning, the model was used to classify sentiment in the LABR dataset. Similarly, AraBERT, which is built on Google's BERT-Base architecture, was pre-trained on about 8.6 billion Arabic words from OSCAR [27], Arabic Wikipedia [28], OSIAN [33], 1.5B words Arabic Corpus [34], and Assafir news. We also fine-tuned AraBERT using the Arabic Sentiment Twitter Corpus [29] to adapt it for sentiment classification. The goal of this phase is to extract sentiment scores from user reviews in the LABR dataset.

3) Combined sentiment score with user rating score: we generate a new adjusted rating that reflects both the explicit user rating and the sentiment score extracted from the written review. Instead of relying only on the numerical rating, our method integrates it with the sentiment score produced by the sentiment analysis model. The adjusted rating is computed as follows [11]:

$$Adjusted\ Rating = \frac{(LABR\ rating + Sentiment\ rating)}{(Scale\ of\ LABR\ rating + Scale\ of\ sentiment\ rating)} \times 5 \quad (1)$$

where, the LABR rating is the original rating on a 5-point scale, and the sentiment rating is derived from the review text on a 3-point scale.

The two components are combined by normalizing their sum using the total scale and then rescaling the result to a 5-point system. This produces an adjusted rating that reflects both the explicit rating and the sentiment expressed in the review. For example, if a user assigns a rating of 5 (on a 5-point scale), and the sentiment analysis produces a score of 1 (on a 3-point scale), the adjusted rating is calculated as:

$$\frac{5+1}{5+3} \times 5 = 3.75$$

The formula assigns equal importance to both components, motivated by their complementary roles: the numerical rating captures explicit user feedback, while the sentiment score reflects implicit opinion expressed in text. Since there is no clear evidence that one component is more important than the other, both are assigned equal weight, providing a balanced baseline. Although alternative weighting schemes could be explored, this study adopts equal weighting, leaving adaptive weighting strategies for future work.

4) Deep learning-based collaborative recommender systems: the main goal of this phase is to use the adjusted ratings from the previous step, which combines sentiment analysis of user reviews and user ratings from the LABR dataset to enhance the recommendations. We developed two deep learning-based collaborative recommender systems: one using a Recurrent Neural Network (RNN) and the other using a Long Short-Term Memory (LSTM) network to learn sequential patterns in user–item interactions. In deep learning, we used "Keras" and "TensorFlow" libraries to develop deep learning model architectures. Additionally, we compared the two deep learning-based collaborative recommender systems using user ratings only and user ratings with sentiment analysis of user reviews. Also, we compare and evaluate the performance of these methods with memory-based collaborative recommender systems.

IV. Results and Discussion

A. Evaluation Metrics

Our proposed models were evaluated by using rating prediction, which measures how close the recommender's estimated ratings are to the actual user ratings. We used two metrics for evaluating the accuracy of predicted ratings: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) [35]. The advantage of using RMSE over MAE is that it penalizes the error more when the difference is high (note that RMSE is always greater than or equal to MAE).

Root Mean Squared Error (RMSE) is a widely used metric because it penalizes large errors when the predicted rating is far from the actual value and less when the predictions are reasonably close. The formula for calculating RMSE is shown below.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \quad (2)$$

where, y_i = actual value, $\widehat{y_i}$ = predicted value, n = number of observations.

Mean Absolute Error (MAE) is the average absolute difference between the predicted and actual ratings. The objective is to minimize the value of MAE. The formula for calculating MAE is shown below:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \quad (3)$$

here, y_i = actual value, $\widehat{y_i}$ = predicted value, n = number of observations

B. Experimental Results

We compared the results of the memory-based and deep learning-based collaborative filtering algorithms, which are types of model-based collaborative filtering algorithms.

1) Memory-based models: We computed both user-user and item-item similarities using two common methods, Cosine similarity and Euclidean distance. Cosine similarity measures the angle between user preference vectors, whereas Euclidean distance measures the geometric distance between them in the feature space. These techniques enable the system to generate recommendations by identifying users or items with similar behavior or preference patterns. The performance of these memory-based models is summarized in Table III.

TABLE III. MEMORY-BASED MODELS RESULTS

Models	Similarity metrics	
	Cosine similarity	*Euclidean distance*
User-User	When K= 25 RMSE = 1.86 MAE = 3.48	When K=75 RMSE = 1.90 MAE = 3.62
Item-Item	**When K= 25** **RMSE = 1.15** **MAE = 1.33**	When K=125 RMSE = 1.18 MAE = 1.39

We found that item-based models perform better and faster than user-based models when we compared the results of the two models. Moreover, Cosine similarity gave better results because it focuses on shared rating patterns between users or items and is less sensitive to rating scale differences, whereas Euclidean distance takes all dimensions into account.

2) Deep learning-based models: We developed two deep learning-based collaborative recommender systems: one using RNN and the other using LSTM. Table IV shows the performance of deep learning-based recommendation systems. As presented in Table IV, the LSTM model outperformed the RNN model as well as the memory-based collaborative filtering models.

TABLE IV. DEEP LEARNING-BASED MODELS RESULTS

Models	Performance
RNN	RMSE = 1.14 MAE = 0.92
LSTM	**RMSE = 1.07** **MAE = 0.88**

3) Memory-based models with Arabic BERT–Mini model and AraBERT: We combine sentiment analysis of user reviews with traditional memory-based recommenders. The Item-Item model with Cosine similarity metrics gave better results than the User-based model when combined with the *Arabic BERT–Mini* model as presented in Table V.

TABLE V. MEMORY-BASED WITH ARABIC BERT–MINI MODEL RESULTS

Models	Cosine similarity
User-User with Arabic BERT–mini model	When K= 25 RMSE = 1.88 MAE = 3.56
Item-Item with Arabic BERT–mini model	**When K= 25** **RMSE = 1.14** **MAE = 1.30**

Also, the Item-Item model with Cosine similarity metrics outperformed the User-based model when combined with the AraBERT model as presented in Table VI.

TABLE VI. MEMORY-BASED WITH ARABERT MODEL RESULTS

Models	Cosine similarity
User-User with AraBERT model	When K= 25 RMSE = 1.90 MAE = 3.63
Item-Item with AraBERT model	**When K= 25** **RMSE = 1.15** **MAE = 1.34**

4) Deep learning-based models with Arabic BERT–Mini model and AraBERT: We combine sentiment analysis of user reviews with deep learning-based recommender systems. The RNN achieved better results than LSTM when combined with the Arabic BERT–Mini model as presented in Table VII.

TABLE VII. DEEP LEARNING-BASED MODELS WITH ARABIC BERT–MINI MODEL RESULTS

Models	Performance
RNN with Arabic BERT–mini model	**RMSE = 0.89** **MAE = 0.83**
LSTM with Arabic BERT–mini model	RMSE = 1.00 MAE = 0.89

Additionally, the RNN model integrated with AraBERT outperforms the LSTM model with AraBERT, as presented in Table VIII.

TABLE VIII. DEEP LEARNING-BASED MODELS WITH ARABERT MODEL RESULTS

Models	Performance
RNN with AraBERT model	**RMSE = 0.89** **MAE = 0.83**
LSTM with AraBERT model	RMSE = 1.02 MAE = 0.83

C. Discussion

The results show the usefulness of integrating the sentiment analysis of user reviews into the recommender system to produce better recommendations. Our proposed approach takes into account two aspects: rating and sentiment analysis, which lead to better recommendations. In addition, the experimental results demonstrate that deep learning-based approaches outperform traditional memory-based models. In particular, the RNN and LSTM models achieved lower RMSE and MAE values compared to User-User and Item-Item methods, confirming their ability to capture complex user-item interactions. Among the deep learning models, RNN showed slightly better performance than LSTM, especially when integrated with Arabic BERT–mini and AraBERT models, where it achieved the best results (RMSE=0.89, MAE=0.83). This improvement highlights the effectiveness of incorporating contextual semantic features derived from sentiment analysis of user reviews. Furthermore, the integration of sentiment analysis significantly enhanced the recommendation accuracy, demonstrating that leveraging textual review information provides richer user preference representation compared to rating-only approaches. However, despite their superior performance, deep learning models require large datasets and substantial computational resources to achieve optimal results. This limitation may affect their applicability due to limited data availability.

Compared to traditional recommendation approaches, such as those reported in [13], that rely primarily on numerical ratings and do not incorporate sentiment information from user reviews, our proposed models achieved lower error rates, particularly when integrating sentiment-aware features. This confirms that incorporating semantic information from user reviews provides a significant advantage over rating-based methods alone.

This study extends our previous work [11] by evaluating RNN and LSTM-based models. However, it does not include a direct comparison with traditional collaborative filtering methods, such as matrix factorization or neighborhood-based approaches. In our previous work [11], we evaluated conventional filtering methods, including SVD, KNN, and NMF, and achieved better performance, as indicated by lower RSME, using the same dataset and sentiment analysis approach. This can be attributed to the fact that deep learning models typically require large amounts of data for effective training, whereas the LABR dataset contains a limited number of reviews per user. In general, both studies confirm that incorporating sentiment analysis of user reviews improves recommendation quality. However, deep learning techniques often require large datasets and significant computational resources to achieve good performance. Additionally, the lack of large-scale Arabic datasets remains a challenge, and future work should focus on creating new datasets to support the development of effective recommender systems for Arabic content.

V. CONCLUSION

We proposed a deep learning-based recommender system enhanced with sentiment analysis of Arabic user reviews. The results show that deep learning models outperform traditional memory-based approaches, with the RNN integrated with Arabic BERT–mini and AraBERT models achieving the best performance. These findings highlight the effectiveness of combining semantic information with ratings to improve accuracy and personalization in recommender systems. The main contribution of this work is the integration of sentiment-aware features into a deep learning-based recommender system for Arabic content, providing a more comprehensive representation of user preferences. Despite these improvements, the reliance on large datasets and high computational resources remains a limitation. Future work will focus on developing more efficient models and evaluating them on larger datasets. Most existing recommender systems research has focused on English or other common languages. However, Arabic presents additional challenges due to its complex grammar and rich vocabulary. Therefore, further research and better support for Arabic datasets are needed to improve recommendation quality.

REFERENCES

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM computing surveys (CSUR), vol. 52, no. 1, pp. 1–38, 2019.

[2] A. Al-Ajlan, S. Alabdulwahab, L. Aljeraisy, A. Althakafi, and R. Alhassoun, “Tourism Group Recommender System,” International Journal of Computer Science and Network Security, vol. 2022, no. 4, pp. 637–644, 2022.

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

[4] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, “Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data,” Expert Systems with Applications, vol. 149, p. 113248, 2020.

[5] B. Alhijawi and Y. Kilani, “A collaborative filtering recommender system using genetic algorithm,” Information Processing & Management, vol. 57, no. 6, p. 102310, 2020.

[6] R. Mu, “A survey of recommender systems based on deep learning,” Ieee Access, vol. 6, pp. 69009–69022, 2018.

[7] S. Lee and D. Kim, “Deep learning based recommender system using cross convolutional filters,” Information Sciences, vol. 592, pp. 112–122, 2022.

[8] C. N. Dang, M. N. Moreno-García, and F. D. la Prieta, “An approach to integrating sentiment analysis into recommender systems,” Sensors, vol. 21, no. 16, p. 5666, 2021.

[9] N. Darraz, I. Karabila, A. El-Ansari, N. Alami, and M. El Mallahi, “Integrated sentiment analysis with BERT for enhanced hybrid recommendation systems,” Expert Systems With Applications, vol. 261, p. 125533, 2025.

[10] S. Nabil, J. El Bouhdidi, and M. Y. Chkouri, “Enhancing Recommender Systems Using Sentiment Analysis: Addressing Cold Start Issues and Improving Recommendation Quality,” vol. 16, no. 4, 2025.

[11] A. Al-Ajlan and N. Alshareef, “Recommender system for arabic content using sentiment analysis of user reviews,” Electronics, vol. 12, no. 13, p. 2785, 2023.

[12] A. Al-Ajlan and N. AlShareef, “A Survey on Recommender System for Arabic Content,” in 2022 5th International Conference on Computing and Informatics (ICCI), New Cairo, Cairo, Egypt: IEEE, Mar. 2022, pp. 316–320. doi: 10.1109/ICCI54321.2022.9756112.

[13] R. Sallam, M. Hussein, and H. Mousa, “An Enhanced Collaborative Filtering-based Approach for Recommender Systems,” International Journal of Computer Applications, vol. 176, no. 41, Jul. 2020.

[14] F. Harrag, A. S. Al-Salman, and A. Alquahtani, “Arabic Opinion Mining Using a Hybrid Recommender System Approach,” arXiv:2009.07397 [cs], Sep. 2020, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/2009.07397

[15] M. Srifi, A. Oussous, A. Ait Lahcen, and S. Mouline, “Evaluation of recent advances in recommender systems on Arabic content,” J Big Data, vol. 8, no. 1, p. 35, Feb. 2021, doi: 10.1186/s40537-021-00420-2.

[16] S. Li, J. Kawale, and Y. Fu, “Deep Collaborative Filtering via Marginalized Denoising Auto-encoder,” in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, in CIKM ’15. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 811–820. doi: 10.1145/2806416.2806527.

[17] A. Liu and J. Gao, “Book recommendation algorithm based on deep learning,” International Journal of Science, vol. 6, no. 10, pp. 152–156, 2019.

[18] C.-H. Lin and H. Chi, “A novel movie recommendation system based on collaborative filtering and neural networks,” in International Conference on Advanced Information Networking and Applications, Springer, 2019, pp. 895–903.

[19] G. S. Aujla et al., “DLRS: deep learning-based recommender system for smart healthcare ecosystem,” in ICC 2019-2019 IEEE international conference on communications (ICC), IEEE, 2019, pp. 1–6.

[20] M. Aly and A. Atiya, “LABR: A Large Scale Arabic Book Reviews Dataset,” in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, Aug. 2013, pp. 494–498.

[21] “pandas - Python Data Analysis Library.” Accessed: Aug. 12, 2025. [Online]. Available: https://pandas.pydata.org/

[22] “re — Regular expression operations — Python 3.10.4 documentation.” Accessed: Aug. 12, 2025. [Online]. Available: https://docs.python.org/3/library/re.html

[23] W. Antoun, F. Baly, and H. Hajj, “Arabert: Transformer-based model for arabic language understanding,” arXiv preprint arXiv:2003.00104, 2020.

[24] A. Safaya, M. Abdullatif, and D. Yuret, “KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media,” in Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 2054–2059.

[25] “AraBERTv2 / AraGPT2 / AraELECTRA. AUB MIND.” Accessed: Aug. 12, 2025. [Online]. Available: https://github.com/aub-mind/arabert

[26] “asafaya bert mini arabic model.” Accessed: Sep. 12, 2025. [Online]. Available: https://metatext.io/models/asafaya-bert-mini-arabic

[27] “OSCAR.” Accessed: Sep. 12, 2025. [Online]. Available: OSCAR

[28] “Wikimedia Downloads.” Accessed: Sep. 12, 2025. [Online]. Available: https://dumps.wikimedia.org/backup-index.html

[29] “Arabic Sentiment Twitter Corpus.” Accessed: Sep. 12, 2025. [Online]. Available: https://www.kaggle.com/datasets/mksaad/arabic-sentiment-twitter-corpus

[30] “Arabic Sentiment Analysis Dataset SS2030 Dataset.” Accessed: Sep. 12, 2025. [Online]. Available: https://www.kaggle.com/snalyami3/arabic-sentiment-analysis-dataset-ss2030-dataset

[31] “Arabic 100k Reviews.” Accessed: Sep. 12, 2025. [Online]. Available: https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews

[32] A. Elmadany, H. Mubarak, and W. Magdy, “Arsas: An arabic speech-act and sentiment corpus of tweets,” Osact, vol. 3, p. 20, 2018.

[33] I. Zeroual, D. Goldhahn, T. Eckart, and A. Lakhouaja, “OSIAN: Open source international Arabic news corpus-preparation and integration into the CLARIN-infrastructure,” in Proceedings of the fourth arabic natural language processing workshop, 2019, pp. 175–182.

[34] I. A. El-Khair, “1.5 billion words arabic corpus,” arXiv preprint arXiv:1611.04033, 2016.

[35] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender systems: an introduction. Cambridge university press, 2010.