electronics

MDPI

Article

Using Hybrid Deep Learning Models of Sentiment Analysis and Item Genres in Recommender Systems for Streaming Services

Cach N. Dang [1,2,3,*], María N. Moreno-García [2] and Fernando De la Prieta [3]

[1] Department of Information Technology, HoChiMinh City University of Transport (UT-HCMC), Ho Chi Minh 70000, Vietnam
[2] Data Mining (MIDA) Research Group, University of Salamanca, 37007 Salamanca, Spain; mmg@usal.es
[3] Biotechnology, Intelligent Systems and Educational Technology (BISITE) Research Group, University of Salamanca, 37007 Salamanca, Spain; fer@usal.es
* Correspondence: cach@ut.edu.vn; Tel.: +84-(91)-9101086

Abstract: Recommender systems are being used in streaming service platforms to provide users with personalized suggestions to increase user satisfaction. These recommendations are primarily based on data about the interaction of users with the system; however, other information from the large amounts of media data can be exploited to improve their reliability. In the case of media social data, sentiment analysis of the opinions expressed by users, together with properties of the items they consume, can help gain a better understanding of their preferences. In this study, we present a recommendation approach that integrates sentiment analysis and genre-based similarity in collaborative filtering methods. The proposal involves the use of BERT for genre preprocessing and feature extraction, as well as hybrid deep learning models, for sentiment analysis of user reviews. The approach was evaluated on popular public movie datasets. The experimental results show that the proposed approach significantly improves the recommender system performance.

Keywords: sentiment analysis; deep learning; genre similarity; streaming services recommendation; natural language processing

check for updates

Citation: Dang, C.N.; Moreno-García, M.N.; De la Prieta, F. Using Hybrid Deep Learning Models of Sentiment Analysis and Item Genres in Recommender Systems for Streaming Services. *Electronics* **2021**, *10*, 2459. https://doi.org/10.3390/electronics10202459

Academic Editors: Marco Lombardi, Domenico Santaniello, Zilu Liang, Muhammad Khan, Arkaitz Zubiaga, George Angelos Papadopoulos and Taeshik Shon

Received: 2 September 2021
Accepted: 7 October 2021
Published: 10 October 2021

1. Introduction

Recommender systems are currently being applied in streaming services platforms to help consumers and the media industry with the discovery and delivery of streaming services. The personalized distribution of streaming services requires the analysis of the item listening/watching behavior by the user; however, other user and item information may also be useful. Collaborative filtering methods are widely used for recommendation in this area. They provide recommendations based on the ratings that users give to items [1].

These techniques yield very good results; however, the difficulty in obtaining explicit feedback in the form of ratings from the users causes the sparsity problem, which occurs when the number of available ratings for the items to be recommended is small. This is the main drawback for the application of this approach in many recommender systems, and in particular in the application domain under study in this work. One way to address this problem is to derive implicit ratings from user behavior, in binary form, based on the existence or not of interaction with the system [2], such as a purchase, or multivalued [3], which requires the analysis of other types of behavior, such as frequency of song playback. When obtaining implicit ratings, other factors can also be taken into account. These include the evolution of user preferences over time [4] and other temporal aspects [5] or the position of the items in the sessions [6]. Appreciating user preferences and behavior can assist to propose a reasonable recommendation to a specific user.

Another source of feedback from users used to infer implicit ratings is the text of their reviews about the items. Deep learning techniques are significant for sentiment analysis on social media comments, thoughts, or feedbacks [7]. Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), or hybrid models are widely used for achieving the highest performance on sentiment analysis tasks [8,9]. Kastrati et al. also applied deep learning techniques for sentiment analysis on students' feedback [10]. Sentiment analysis of these texts is a helpful tool for inferring user preferences and use them in recommender systems. Some examples of this can be found in the work of Dang et al. [11], in which two hybrid deep learning models were applied to analyze sentiment in reviews. The output was used to improve and validate the recommendations of a recommender system. Kumar et al. [12] proposed a hybrid recommender system by combining collaborative filtering and content-based filtering with the use of sentiment analysis of movie tweets to boost up the recommender system. The problem of automatically extracting opinions from online users has been a growing research topic recently [7]. Social media data have been exploited in different ways to address some problems, especially associated with collaborative filtering approaches [13]. In addition, Rosa et al. [14] used a sentiment intensity metric to build a music recommender system. Users' sentiments are extracted from sentences posted on social networks and the recommendations are made using a framework of low complexity that suggests songs based on the current user's sentiment intensity. The research by Osman et al. [15] addressed the data-sparsity problem of recommender systems by integrating a sentiment-based analysis. Their work was applied to the Internet Movie Dataset (IMDb) and Movie Lens datasets, but improvements in sentiment analysis have been made since the paper was published. In particular, when only sparse rating data are available, sentiment analysis can play a key role in improving the quality of recommendations. This is because recommendation algorithms mostly rely on users' ratings to select the items to recommend. Such ratings are usually insufficient and very limited. On the other hand, sentiment-based ratings of items that can be derived from reviews or opinions given through online news services, blogs, social media, or even the recommender systems themselves are seen as being capable of providing better recommendations to users.

In addition, some recommendation approaches leverage item metadata to deal with problems mainly associated with collaborative filtering methods [16]. Among such data, social tags have become an important input to recommender systems for streaming platforms. Many efforts have been addressed to unify tagging information to reveal behavior and extract the latent semantic relations among items [17]. In Reference [18], the authors proposed a method for automatic generation of social tags for music recommendation. The purpose is to avoid the cold-start problem common in such systems, when a user or an item is newly added to the system and as a result has few ratings. Instead of relying on ratings in a music recommendation method [1,3], social tags may be used to improve music recommender systems by calculating the similarity between music pieces by combining both tag and rating [13,19], in the same way that other item attributes, such as movie genres or music audio features, are used to classify items or establish item similarity [18,19]. In Reference [20], musical genre classification was performed according to spectrum, rhythm, and harmony. Audio features and tags were used in Reference [21], where a method for recommending appropriate music for videos was presented. Videos and music items were represented as a linear combination of latent factors related to their features. Low-level description of the music was also used in Reference [22] for emotion recognition and genre classification.

Social tag embedding also was used in a collaborative filtering approach in which user similarities based on both tag embedding and ratings were combined to generate the recommendations in Reference [13].

Sentiment-based models on reviews and tags have been exploited in recommender systems to overcome the data-sparsity problem that exists in conventional recommender systems. However, either the tags or the reviews are not available in some streaming platforms; thus, they cannot be used together. Some datasets, such as Amazon music, have ratings and reviews without social tags, while datasets from lasf.fm [23] or MusicBrainz [24] have tags but no reviews. To overcome this problem, we can resort to the genre attribute, which characterizes the items as tags do, and which is present in most of the datasets. This and other attributes have been commonly used in content-based methods to recommend items similar to those that the user has previously consumed or rated positively. In Reference [25], genres that the user might prefer to watch on Movie Lens dataset were used to provide the best suggestions possible. Gunawan et al. [26] presented a work in which genres were predicted by a model of convolutional recurrent neural networks applied for recommendation. In some works, tags were used to predict movie or music genres; thus, depending on the purpose of their use, in some recommender systems where genres are available, these could be used instead of tags. In fact, many social tag values from last.fm or MusicBrainz are really similar to the genres of artists. An example of this is the work of Hong et al. [27], who proposed a tag-based method to calculate similarities between artists and then classify them into genres with the k-NN algorithm on the laft.fm database.

Our study raises whether integrating sentiment analysis and embedding of item attributes such as genres in recommender systems may significantly enhance the recommendation quality. In this study, we proposed to take advantage of the genre attribute and hybrid deep-learning-based sentiment analysis of reviews to improve collaborative filtering-based recommender systems in the realm of streaming services. The difference with other works in the literature lies in the fact that genre is not used in the context of content-based methods or to obtain similarity between items but to characterize users and thus provide better recommendations. Moreover, this attribute is not used raw as in most recommendation methods but is preprocessed with advanced natural-language-processing techniques. Regarding sentiment analysis, the proposed approach incorporates new specific techniques for feature extraction and hybrid deep learning methods.

The main contribution of our work to the literature lies in the proposal of new hybrid deep learning methods for sentiment analysis and their incorporation into recommender systems based on collaborative filtering, as well as the use of an item attribute, previously preprocessed with NLP techniques, to characterize the users.

The rest of this paper is organized as follows. Section 2 provides the description of the material and methods used in this study. Section 3 shows the results given by our proposal and their comparison with the baseline results. Section 4 outlines the discussion, and Section 5 offers the main conclusions.

2. Material and Methods

2.1. Collection of Data

Data gathering is the first requirement for any recommendation model. There are two categories of data that are collected: implicit or explicit. Implicit data include customer's actions via such as order history, return history, page view, etc. Meanwhile, explicit data contain user's actions online via the internet, including ratings, reviews of movies/songs, etc. In this study, we chose the datasets based on availability and accessibility criteria. Moreover, we took into account that they are widely accepted by the research community. The datasets used in the study to validate our proposal are described below.

- Multimodal Album Reviews Dataset (MARD) [28] contains text and metadata, which are retrieved from Amazon customer-review datasets. The music metadata of this dataset are enriched by MusicBrainz, and the audio description is updated with AcousticBrainz. In total, MARD stores 65,566 albums and 263,525 customer reviews.
- Amazon Movie Reviews consists of movie reviews from Amazon [29]. Each review also includes product and user information, ratings, and plaintext reviews. It covers a period of more than 10 years, as well, including 7,911,684 reviews with 889,176 users and 253,059 products up to October 2012.

The dataset MARD was built by combining two files, mard_metadata.json and mard_reviews.json. From the mard_reviews.json, we collected reviewID, itemID, review, and rating. Then, through itemID, we map to mard_metadata.json to get more information, including genre of the album and artist. Total data has 263,525 samples. Rating values are from 1 to 5.

The second dataset used in our study is named Amazon Movie. This dataset consists of movie reviews from the Amazon Movie Reviews dataset. Reviews include product and user information, ratings, and a plaintext review. For each product in the dataset, we crawled genre information from the Amazon system [30] and added it to the dataset. We collected a total of 203,967 samples.

Finally, we completed two datasets with users, ratings, reviews, and genres. Figure 1 visualizes the word cloud of these datasets, one related to movies and the other related to music.

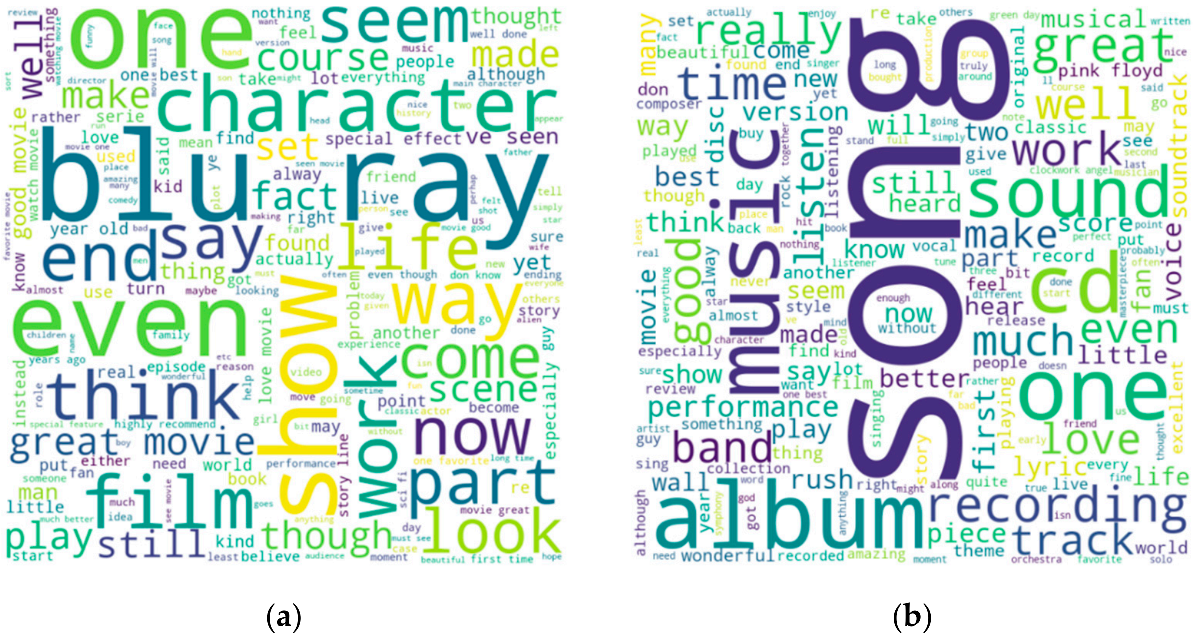


Figure 1. Word cloud of the combined dataset: (**a**) Amazon Movie dataset in the first panel; (**b**) MARD (Multimodal Album Reviews Dataset) dataset in the second panel.

2.2. *Proposed Recommendation Method*

Recommender systems rely on explicit user ratings, but this is not feasible in an increasing number of domains. Moreover, when explicit ratings are available, the trust and reliability of the ratings may limit the recommender system performance. When we have a large number of reviews and the genres on these items, using the last and analyzing the sentiments in the review texts to obtain implicit feedback, in addition to traditional ratings for items, is useful and helps to improve the recommendations to users. In this study, we propose the application of advanced feature extraction techniques and hybrid deep learning methods for sentiment analysis. The advantages of BERT are exploited for both preprocessing genres and feature extraction from reviews as a preliminary step in the deep-learning-based sentiment analysis. The objective is to improve the performance and reliability of recommender systems for streaming platforms. Figure 2 illustrates the architecture of recommender systems for streaming services based on hybrid deep learning models of sentiment analysis and item genres.

BERT is used to create feature vectors. BERT is a language model for Natural Language Processing (NLP) that was published by researchers at Google AI Language in 2018 [31]. A pretrained BERT model was used in this study. The reviews and genre data are fed into the BERT model to generate the feature vectors. In the case of genres, the vectors are used to compute the weight of the user similarity, while feature vector obtained from reviews are the input to the hybrid deep learning models that perform the sentiment classification.

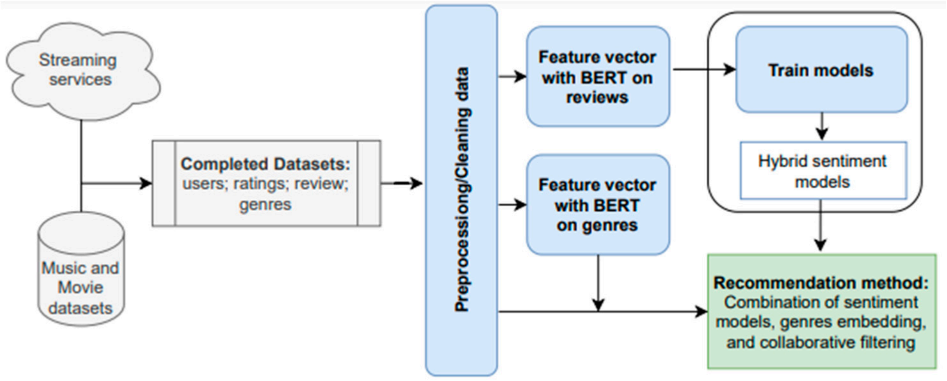

Streaming services
Music and Movie datasets
Completed Datasets: users; ratings; review; genres
Preprocessiong/Cleaning data
Feature vector with BERT on reviews
Feature vector with BERT on genres
Train models
Hybrid sentiment models
Recommendation method: Combination of sentiment models, genres embedding, and collaborative filtering

Figure 2. Architecture of recommender systems for streaming services based on hybrid deep learning models of sentiment analysis and item genres.

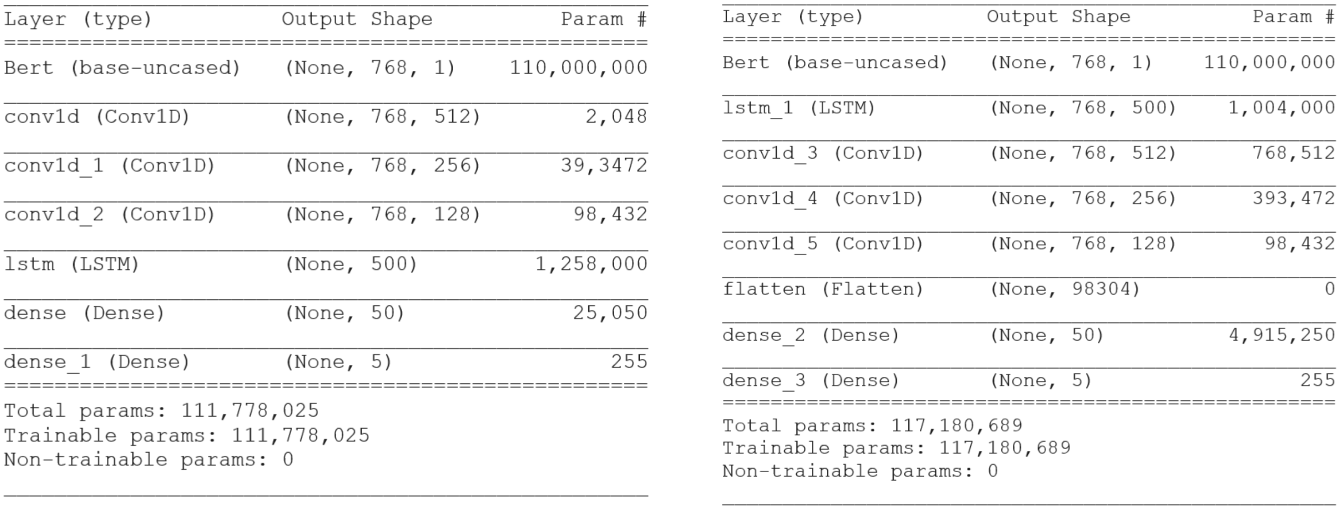
The hybrid models can increase sentiment analysis accuracy compared to a single model performance [9]. Our proposal involves two hybrid deep learning models with variations in using CNN [32] and LSTM [33] networks in the deep learning layers to incorporate the advantages of both and thus fill some shortcomings of individual methods. The combination helped to take advantage of CNN and LSTM: CNN can extract characteristics, and LSTM can store past information at the state nodes. The first hybrid model combines CNN and LSTM, and the second hybrid model combines LSTM and CNN. We labeled the reviews with one value of an ordinal scale of five classes (very negative, negative, neutral, positive, and very positive), analogous to the explicit ratings, to train and validate the result of sentiment analysis. The visualization of these model connections, the connection process, and the data-processing flow are indicated in Figure 3. These models were printed from the code after we conducted and setup these models. Value “None” means that this dimension is variable. The “None” dimension in our model is always the batch size which does not need be fixed. The function embedding is the embedding layer that is initialized with random weights, and which will learn the embedding for all words in the training dataset. Then, the hybrid models combine two popular deep learning models, namely CNN and LSTM [7], and take advantage of the two network architectures when performing sentiment analysis. Finally, the output layer has a Relu activation function.

The proposed recommendation method is a user-based collaborative filtering approach that considers explicit ratings, implicit ratings inferred from reviews’ sentiment analysis as well as user similarity derived from user ratings and item genres previously preprocessed with BERT. The objective is to achieve better predictive accuracy than widely used collaborative filtering (CF) methods, such as Singular Value Decomposition (SVD) [34], Non-Negative Matrix Factorization (NMF) [35], and SVD++ [36]. The proposed method can be applied using these or other CF methods as a basis. Results from the CF recommendation method and sentiment analysis and genres are combined to predict ratings and create a list of recommendations.

The procedure requires us to compute the similarity between the active user, u_a, and his neighbor user, u_i, which would be obtained by using the cosine metric [37], as in Equation (1). In our case, the neighbors of user u_a are users who have rated the same items as user u_a in a similar way or the score of their reviews on the same items are similar.

$$Sim(u_a, u_i) = \frac{\sum_{j=1}^{n} r_{aj}\, r_{ij}}{\sqrt{\sum_{j=1}^{n} r_{aj}^2}\,\sqrt{\sum_{j=1}^{n} r_{ij}^2}} \tag{1}$$

User similarities based on ratings given by Equation (1) are weighted by considering similarities between users in terms of the genre of the items they consume (music, movies, etc.). Therefore, the genres of all items rated by user u_a and u_i are used to determine the weight of the $Sim(u_a, u_i)$. For each item of user u_a, we got the genres and combined them into a string, s_a, and converted s_a to a vector, v_a. Similarly, for each item user, u_i, we also got the genres, combined them into a string (s_i) and converted s_i to a vector (v_i). BERT is used to obtain the u_a and v_i vectors. Since gender are used to characterize the user, each input to the BERT model consists of the genders of all items rated by a given user, u_i. The weight of $Sim(u_a, u_i)$ was determined by the normalized distance between v_a and v_i. We used Euclidean distance [38] to calculate distance between v_a and his neighbor v_i.



```
Layer (type)            Output Shape          Param #
==================================================
Bert (base-uncased)    (None, 768, 1)    110,000,000
__________________________________________________
conv1d (Conv1D)        (None, 768, 512)        2,048
__________________________________________________
conv1d_1 (Conv1D)      (None, 768, 256)      39,3472
__________________________________________________
conv1d_2 (Conv1D)      (None, 768, 128)       98,432
__________________________________________________
lstm (LSTM)            (None, 500)         1,258,000
__________________________________________________
dense (Dense)          (None, 50)             25,050
__________________________________________________
dense_1 (Dense)        (None, 5)                 255
==================================================
Total params: 111,778,025
Trainable params: 111,778,025
Non-trainable params: 0
__________________________________________________
```

(**a**)

```
Layer (type)            Output Shape            Param #
=================================================
Bert (base-uncased)     (None, 768, 1)    110,000,000
_________________________________________________
lstm_1 (LSTM)           (None, 768, 500)    1,004,000
_________________________________________________
conv1d_3 (Conv1D)       (None, 768, 512)      768,512
_________________________________________________
conv1d_4 (Conv1D)       (None, 768, 256)      393,472
_________________________________________________
conv1d_5 (Conv1D)       (None, 768, 128)       98,432
_________________________________________________
flatten (Flatten)       (None, 98304)               0
_________________________________________________
dense_2 (Dense)         (None, 50)          4,915,250
_________________________________________________
dense_3 (Dense)         (None, 5)                 255
=================================================
Total params: 117,180,689
Trainable params: 117,180,689
Non-trainable params: 0
_________________________________________________
```

(**b**)

Figure 3. Visualization of the hybrid models: (**a**) hybrid CNN–LSTM (Convolutional Neural Networks Long Short-Term Memory) model in the first panel; (**b**) hybrid LSTM–CNN model in the second panel.

$Sim(u_a, u_i)$ and the weight $W(u_a, u_i)$ are used in Equation (2) for rating prediction based on user similarity. The ratings of the k most similar users (r_{aj}) are used to estimate the preferences of the active user, u_a, about the item i_j that he/she has not rated.

$$pr_{aj} = \overline{r_a} + \frac{\sum_{i=1}^{K} W(u_a, u_i) \times Sim(u_a, u_i) \times \left(r_{ij} - \overline{r_i}\right)}{\sum_{i=1}^{K} \left| W(u_a, u_i) \cdot Sim(u_a, u_i) \right|} \tag{2}$$

where r_{ij} is the rating that user u_i gives to item i_j respectively; $\overline{r_a}$ and $\overline{r_i}$ are the average ratings of user u_a and user u_i, respectively; and $Sim(u_a, u_i)$ is the similarity between the active user u_a and his neighbor user u_i; $W(u_a, u_i)$ is the weight of $Sim(u_a, u_i)$.

Given a rating matrix $R_{m \times n}$ $(\mathbb{N})$ for training, where m is the number of users and n is the number of items, $r_{ij} \in R_{m \times n}$ denotes the rating of user u_i on item i_j.

These rating predictions are used in the sentiment-based recommendation model whose prediction is denoted by pr_sent_{aj}. The procedure begins with the classification of each item review in one of five possible classes by means of the hybrid deep learning models. Each class is associated with one of the sentiment scores from 1 to 5 to be consistent with rating values. Then, for each user u_a, all items i_k rated by user u_a whose sentiment score matches the explicit rating are found. The next step is to find, for each item i_j, all users who already rated item i_j and item i_k in the training set and their review scores also match the explicit ratings.

The two lists of data, including items and users, which are created in the previous steps, are used for predicting user u_a rating on each item i_j that user u_a has not rated. That prediction denoted as pr_sent_{aj} is obtained by using Equation (2).

After all, the final prediction fpr_{aj} of the rating of user u_a on item i_j in the test set is computed as follows:

$$fpr_{aj} = \beta \times pr\_mf_{aj} + (1 - \beta) \times pr\_sent_{aj} \tag{3}$$

where pr_mf_{aj} is the rating for user u_a and item i_j predicted by Matrix Factorization methods (SVD, SVD++, and NMF), without using sentiments; pr_sent_{aj} is the rating for user u_a and item i_j predicted by using the sentiment model; and β parameter used to adjust the importance of each term of the equation.

2.3. Experimental Setup

We performed experiments with two different settings without/with sentiment analysis and genres. In the former, recommendations are based on recommender system methods without sentiment, while in the second, the result of performing sentiment analysis on the reviews and using genre-based user similarity is incorporated into the recommendation process.

The configuration of related parameters, hardware devices, and the necessary library facilities was carried out before performing the experiments, such as echo = 5 and k-fold = 5. In particular, we used Google Colab Pro with GPU Tesla P100-PCIE-16GB or GPU Tesla V100-SXM2-16GB [39], Keras [40], Pytorch [41], and Surprise libraries. We also used the implementation of the SVD, NMF, and SVD++ algorithms provided by the Surprise library [42].

3. Results

We tested three widely used CF recommendation methods, namely SVD, NMF, and SVD++ [43], as baseline to validate our proposal. In addition, two complete datasets, namely MARD and Amazon Movie, were used in the study. As mentioned previously, we applied two hybrid deep learning models for sentiment analysis: CNN–LSTM and LSTM–CNN, referred to as C-LSTM and L-CNN, respectively. Finally, to validate our recommendation approach, we compared the performance of the CF recommendation algorithms with two different settings, without/with sentiment analysis and genres. CF recommendation methods without incorporating sentiment and genres were used as a baseline. The same techniques were tested with our proposal involving sentiment analysis of reviews and user similarity based on genres (With Sentiment and Genres—WSG).

In this section, we present the results of the experiments conducted to evaluate the performance of the proposed approach for recommender systems in the area of streaming services. As is usual in the field of recommender systems, two types of evaluation were carried out by using specific metrics for each of them. First, we evaluated the top-*n* recommendation lists containing the items with the highest predicted rating values. Secondly, error rates in the prediction of the ratings were computed. Thus, the comparative study was conducted for both item recommendation (recommendation of top-*n* lists) and rating prediction.

3.1. Evaluation of Top-n Recommendations

Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) were used for evaluating top-*n* recommendations setting $n = 5$. The results obtained with the MARD dataset are shown in Tables 1 and 2 and illustrated in Figures 4 and 5, while the results for the Amazon Movie datasets are shown in Tables 3 and 4 and Figures 6 and 7. It can be seen that, in most cases, the metrics provide higher values with the proposal against the baseline regardless of the CF method used.

Table 1. MRR, MAP, and NDCG values (%) without and with L-CNN sentiment and genres-based model on the MARD dataset with different β values.

	MRR			**MAP**			**NDCG**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	85.63	84.22	85.35	74.23	72.34	74.12	87.59	86.85	87.88
WSG (β = 0.3)	84.93	87.20	85.93	74.24	74.79	74.86	86.73	88.21	88.26
WSG (β = 0.5)	85.66	83.57	86.46	74.50	72.68	75.15	87.39	86.89	88.32
WSG (β = 0.7)	86.93	83.51	86.99	75.35	72.30	75.47	88.10	86.69	88.19

Table 2. MRR, MAP, and NDCG values (%) without and with C-LSTM sentiment and genres-based model on the MARD dataset with different β values.

	MRR			**MAP**			**NDCG**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	85.63	84.22	85.35	74.23	72.34	74.12	87.59	86.85	87.88
WSG (β = 0.3)	84.58	87.00	85.65	73.97	74.85	74.76	86.67	88.35	88.27
WSG (β = 0.5)	85.67	83.45	86.26	74.49	72.70	75.03	87.41	86.73	88.20
WSG (β = 0.7)	86.44	83.50	86.70	75.14	72.30	75.33	88.00	86.73	88.23

Table 3. MRR, MAP, and NDCG values (%) without and with L-CNN sentiment and genres-based model on Amazon Movie dataset with different β values.

	MRR			**MAP**			**NDCG**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	84.90	84.57	85.23	73.55	73.61	74.58	87.00	86.86	87.80
WSG (β = 0.3)	86.19	84.60	86.01	75.16	73.83	75.54	87.83	87.16	88.63
WSG (β = 0.5)	86.35	84.54	86.10	75.36	73.98	75.76	88.09	87.38	88.89
WSG (β = 0.7)	86.11	84.51	85.91	75.17	73.92	75.62	87.85	86.92	88.37

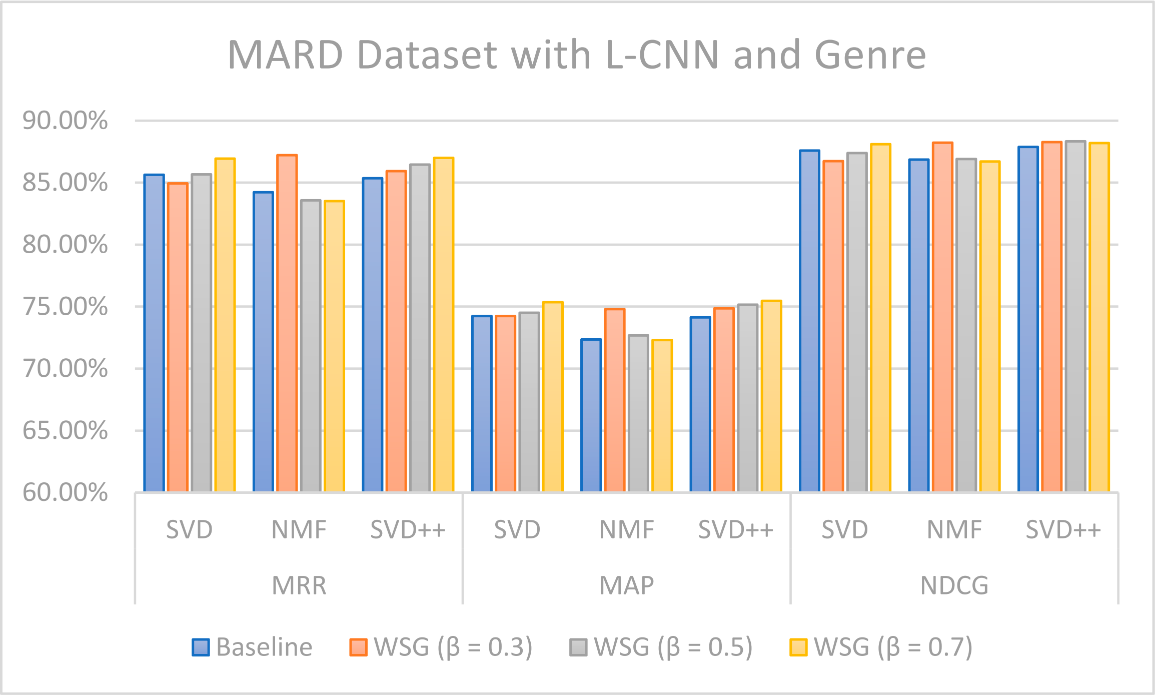


Figure 4. MRR (Mean Reciprocal Rank), MAP (Mean Average Precision), and NDCG (Normalized Discounted Cumulative Gain) values without and with L-CNN (Long Short-Term Memory-Convolutional Neural Networks) sentiment and genres-based model on the MARD dataset with different β values.

Table 4. MRR, MAP, and NDCG values (%) without and with C-LSTM sentiment and genres-based model on Amazon Movie dataset with different β values.

	MRR			**MAP**			**NDCG**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	84.90	84.57	85.23	73.55	73.61	74.58	87.00	86.86	87.80
WSG (β = 0.3)	85.65	85.86	85.33	74.50	75.14	75.02	87.61	87.77	88.04
WSG (β = 0.5)	86.54	85.95	85.24	75.01	75.46	75.14	88.07	87.92	88.40
WSG (β = 0.7)	86.05	85.63	85.33	74.78	75.28	75.02	87.85	87.63	88.05

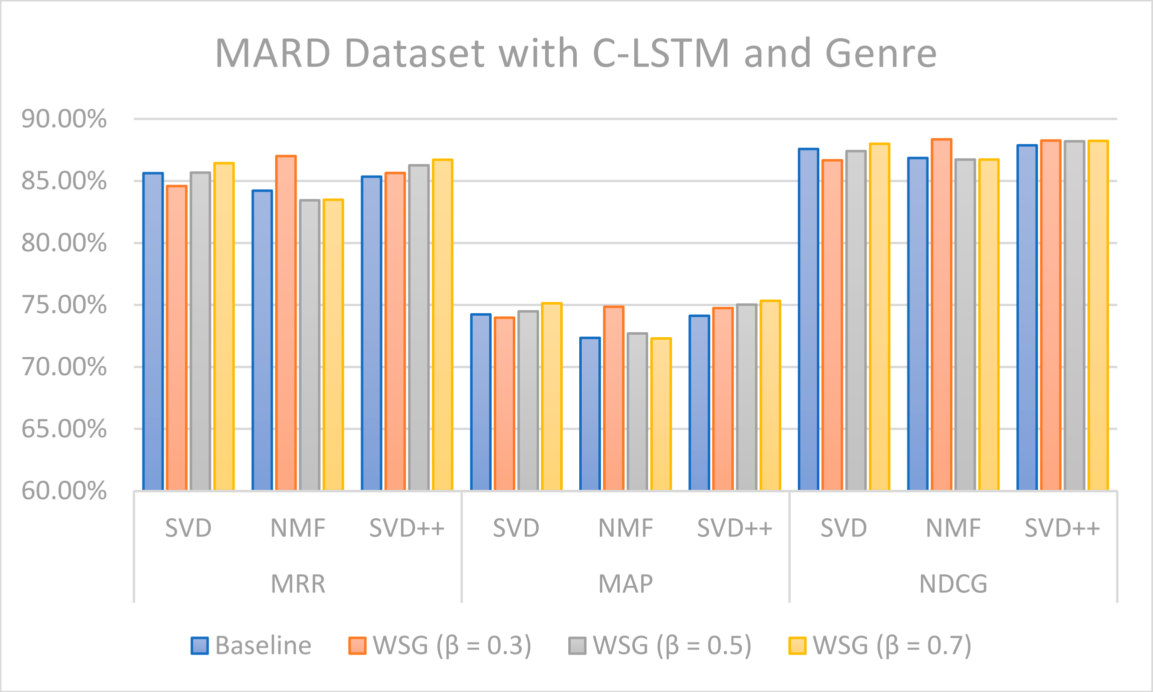


Figure 5. MRR, MAP, and NDCG values without and with C-LSTM sentiment and genres-based model on the MARD dataset with different β values.

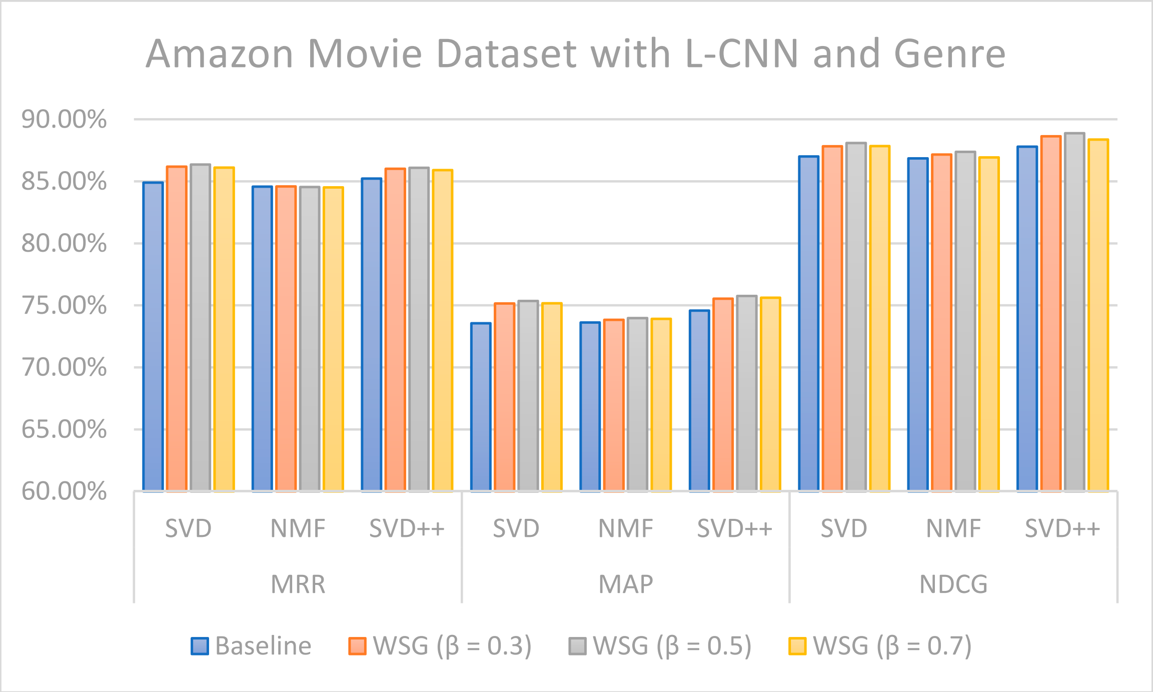


Figure 6. MRR, MAP, and NDCG values without and with L-CNN sentiment and genres-based model on Amazon Movie dataset with different β values.

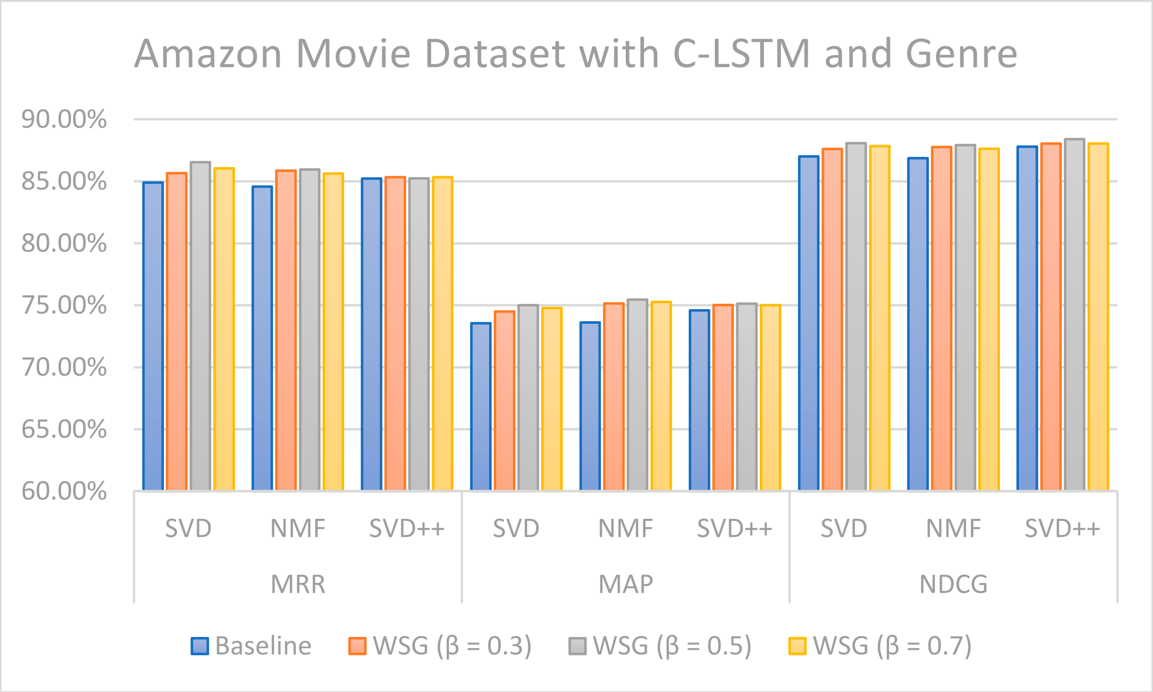


Figure 7. MRR, MAP, and NDCG values with C-LSTM sentiment and genres-based model on Amazon Movie dataset with different β values.

3.2. Evaluation of Rating Prediction

The metrics used to compute error rates in rating predictions were Root-Mean-Square Error (RMSE), Mean Absolute Error (MAE), and Normalized MAE (NMAE). Tables 5–8 show these measures for rating prediction on MARD and Amazon Movie datasets. They were calculated based on the application of SVD, NMF, and SVD++ algorithms with and without using sentiment analysis. Tables 5 and 7 contain the values obtained when using L-CNN hybrid deep learning sentiment models, while Tables 6 and 8 show the results when using the C-LSTM hybrid deep learning sentiment model. Figures 8–11 illustrate the comparative results obtained from the recommendation methods with sentiment analysis with different β parameter values against those obtained from the same methods without sentiment analysis and genres. Figures 12 and 13 illustrate the comparison of the sentiment and genres-based model applied with β = 0.7 against baseline on both datasets.

Table 5. RMSE, MAE, and NMAE values with and without L-CNN sentiment and genres-based model and different values of β on the MARD dataset.

	RMSE			**MAE**			**NMAE**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	0.9193	1.1865	0.9107	0.8307	0.9648	0.8224	0.2077	0.2412	0.2056
WSG (β = 0.3)	0.9285	1.1548	0.9241	0.6844	0.8843	0.6748	0.1711	0.2211	0.1687
WSG (β = 0.5)	0.9190	1.1557	0.9140	0.6839	0.8930	0.6733	0.1710	0.2233	0.1683
WSG (β = 0.7)	0.9149	1.1632	0.9089	0.6850	0.9056	0.6732	0.1712	0.2264	0.1683

Table 6. RMSE, MAE, and NMAE values with and without C-LSTM sentiment and genres-based model and different values of β on the MARD dataset.

	RMSE			**MAE**			**NMAE**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	0.9193	1.1865	0.9107	0.8307	0.9648	0.8224	0.2077	0.2412	0.2056
WSG (β = 0.3)	0.9285	1.1547	0.9241	0.6845	0.8841	0.6749	0.1711	0.2210	0.1687
WSG (β = 0.5)	0.9190	1.1556	0.9140	0.6840	0.8929	0.6734	0.1710	0.2232	0.1683
WSG (β = 0.7)	0.9150	1.1631	0.9089	0.6849	0.9054	0.6733	0.1712	0.2264	0.1683

Table 7. RMSE, MAE, and NMAE values with and without L-CNN sentiment and genres-based model and different values of β on Amazon Movie dataset.

	RMSE			MAE			NMAE		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	1.1685	1.2004	1.1436	0.9581	0.9583	0.9426	0.2395	0.2396	0.2356
WSG (β = 0.3)	1.0793	1.1624	1.0693	0.8109	0.8533	0.7969	0.2027	0.2133	0.1992
WSG (β = 0.5)	1.0919	1.1644	1.0789	0.8373	0.8652	0.8191	0.2093	0.2163	0.2048
WSG (β = 0.7)	1.1151	1.1736	1.0981	0.8675	0.8824	0.8447	0.2169	0.2206	0.2112

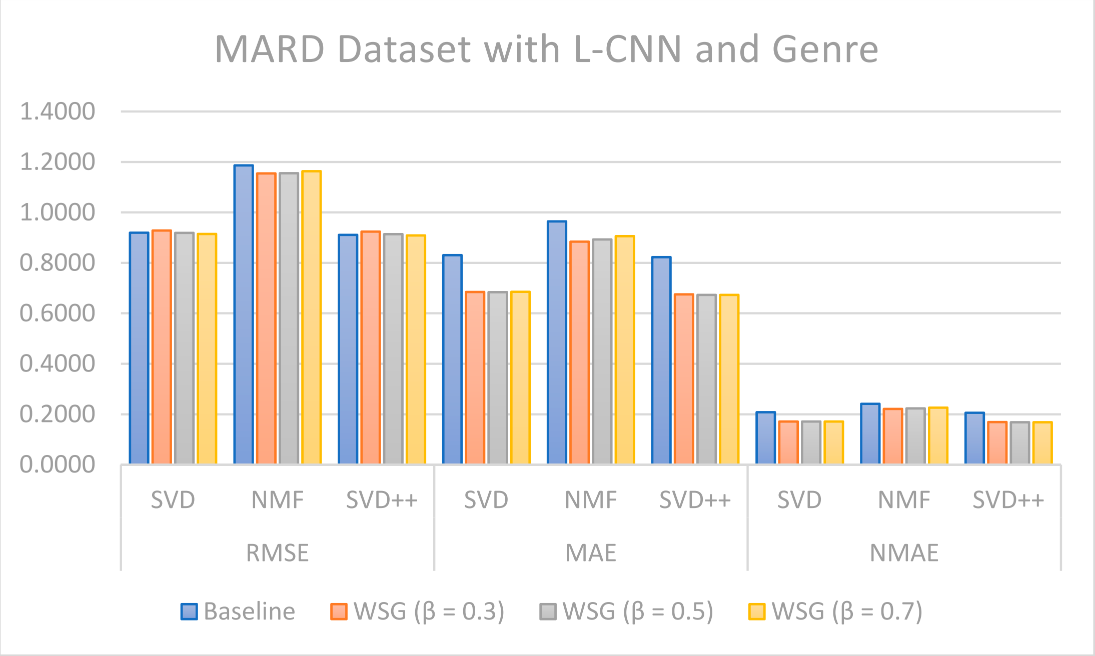


Figure 8. RMSE (Root-Mean-Square Error), MAE (Mean Absolute Error), and NMAE (Normalized MAE) values with and without L-CNN sentiment and genres-based model and different values of β on the MARD dataset.

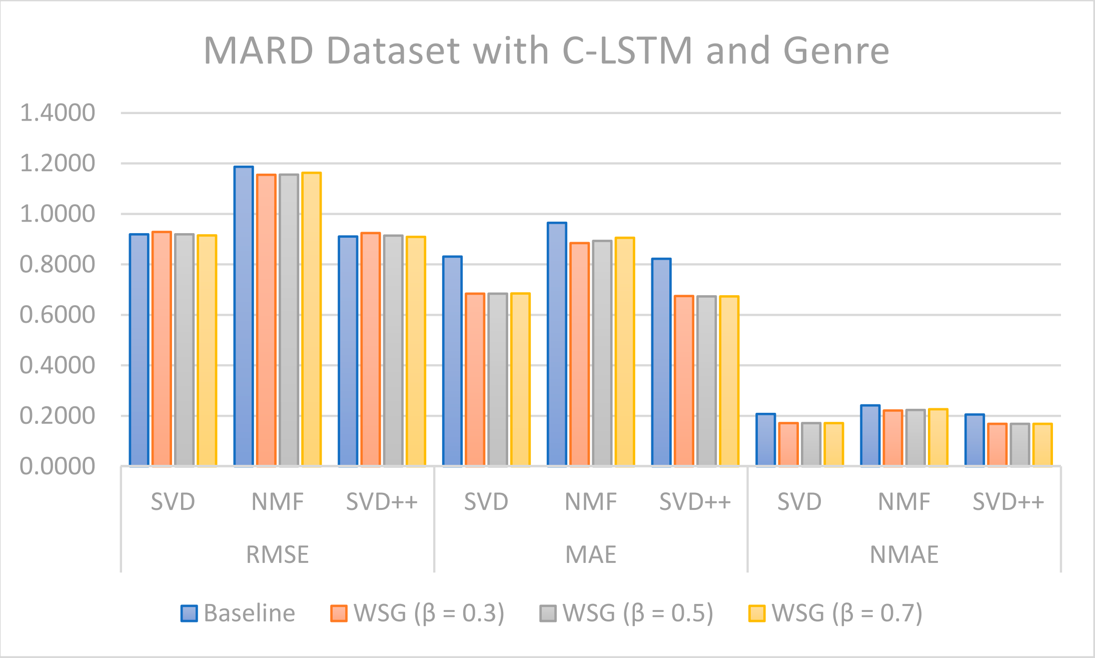


Figure 9. RMSE, MAE, and NMAE values with and without C-LSTM sentiment and genres-based model and different values of β on the MARD dataset.

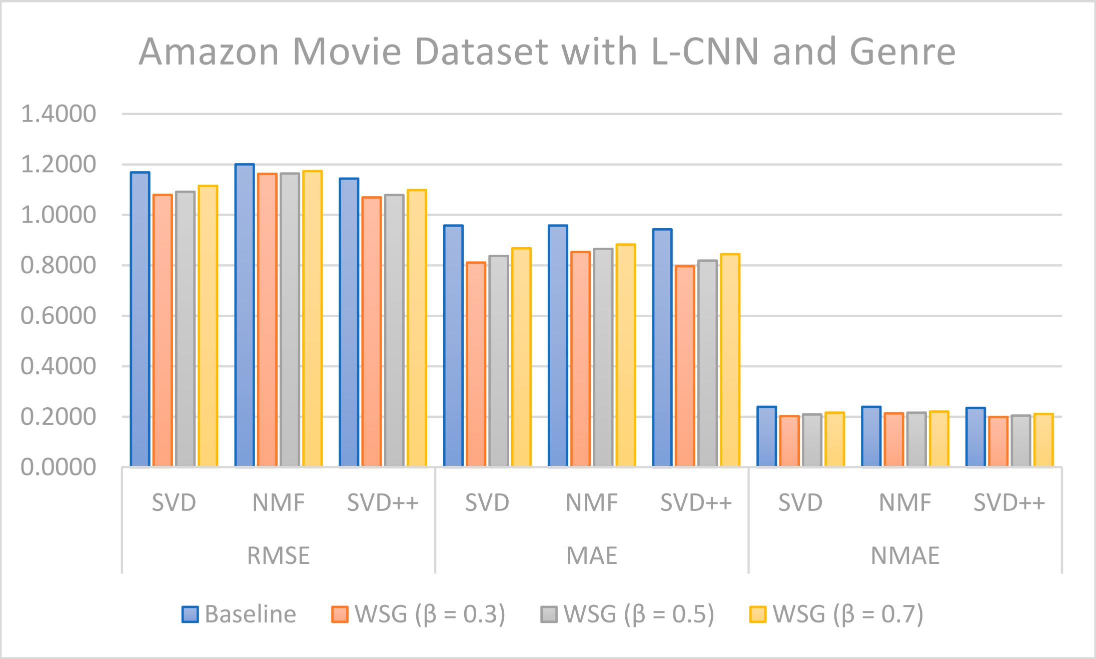


Figure 10. RMSE, MAE, and NMAE with and without L-CNN sentiment and genres-based model and different values of β on Amazon Movie dataset.

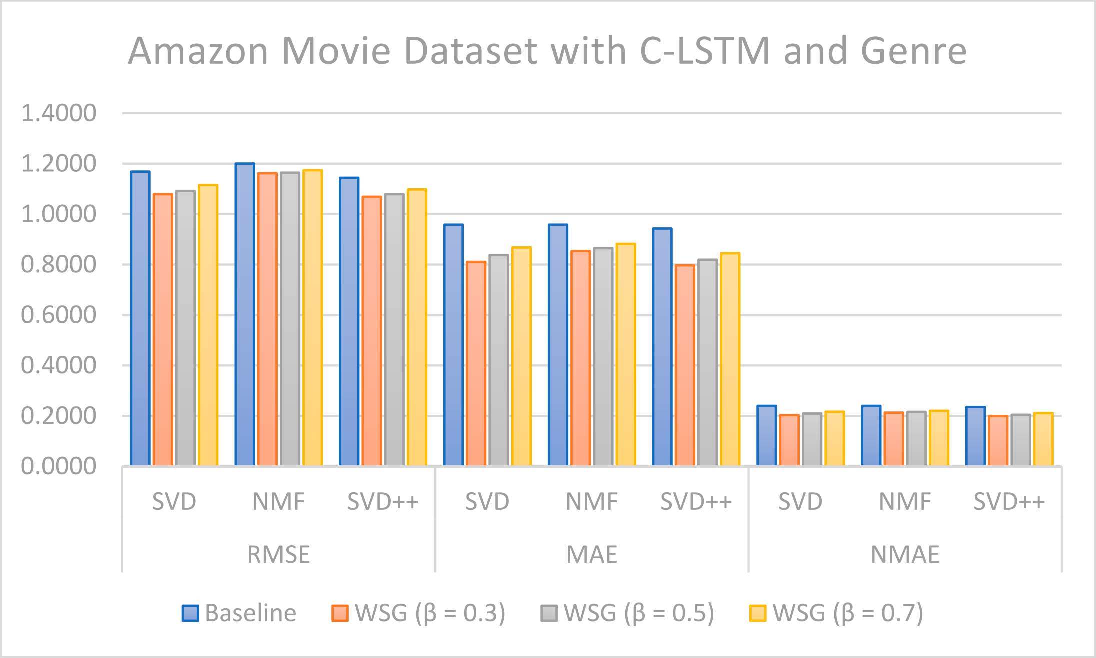
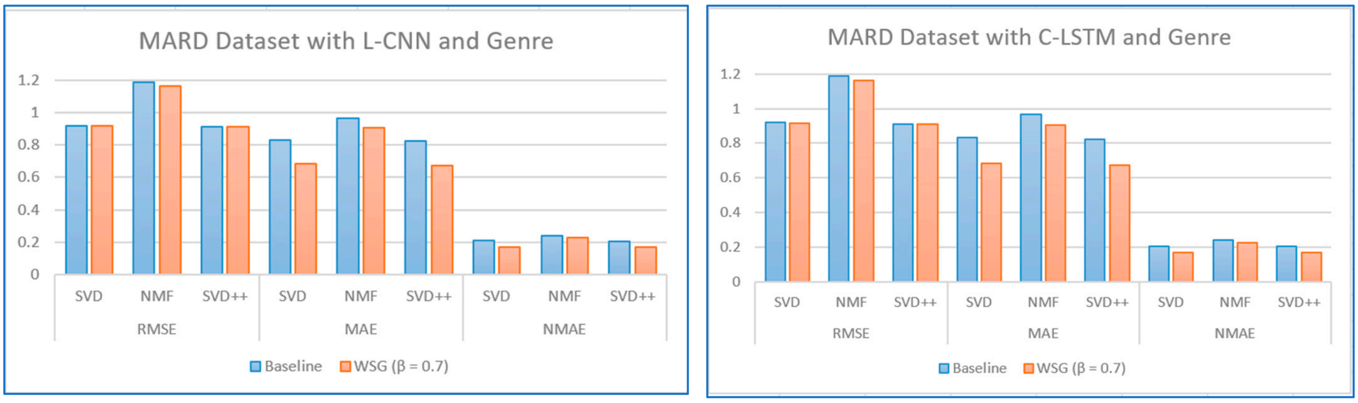


Figure 11. RMSE, MAE, and NMAE with and without C-LSTM sentiment and genres-based model and different values of β on Amazon Movie dataset.

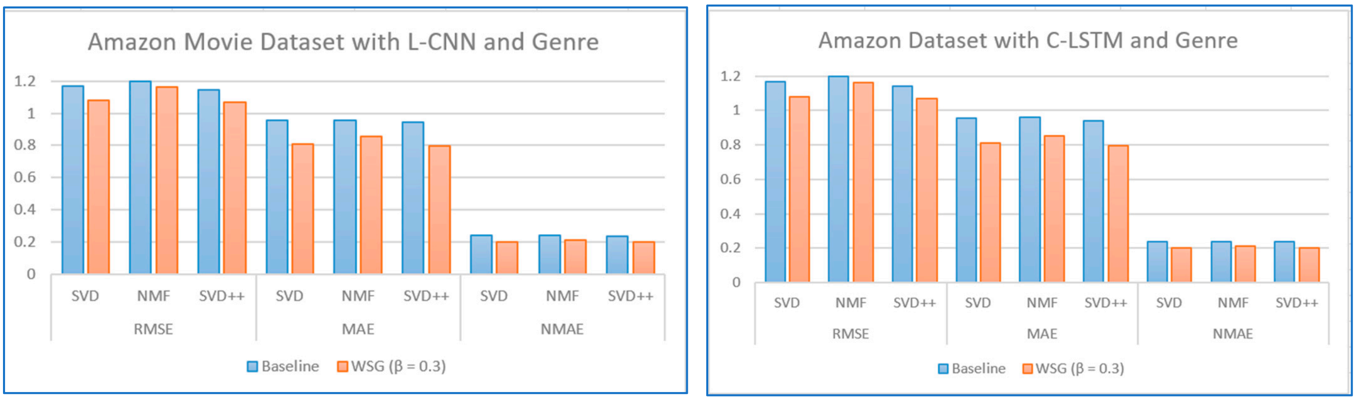
Table 8. RMSE, MAE, and NMAE with and without C-LSTM sentiment and genres-based model and different values of β on Amazon Movie dataset.

	RMSE			**MAE**			**NMAE**		
	SVD	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**	**SVD**	**NMF**	**SVD++**
Baseline	1.1685	1.2004	1.1436	0.9581	0.9583	0.9426	0.2395	0.2396	0.2356
WSG (β = 0.3)	1.0789	1.1621	1.0689	0.8105	0.8530	0.7965	0.2026	0.2133	0.1991
WSG (β = 0.5)	1.0916	1.1643	1.0786	0.8371	0.8650	0.8188	0.2093	0.2162	0.2047
WSG (β = 0.7)	1.1150	1.1736	1.0979	0.8674	0.8823	0.8446	0.2168	0.2206	0.2111



(**a**) (**b**)

Figure 12. Comparison of the sentiment and genres-based model applied with β = 0.7 against baseline on MARD dataset: (**a**) hybrid LSTM–CNN model in the first panel; (**b**) hybrid CNN–LSTM model in the second panel.



(**a**) (**b**)

Figure 13. Comparison of the sentiment and genres-based model applied with β = 0.3 against baseline on Amazon Movie dataset: (**a**) hybrid LSTM–CNN model in the first panel; (**b**) hybrid CNN–LSTM model in the second panel.

4. Discussion

The results shown in the previous section show that the proposal presented in this paper outperforms the baselines both in the evaluation of the top n recommendation lists and in the prediction of ratings.

The values of MRR, MAP, and NDCG show that the proposed methods can improve topN recommendations. In the case of the Amazon Movie dataset for the SVD algorithm combined with L-CNN sentiment and genres-based model with different β values, the increase was 1.46 (MMR), 1.81 (MAP), and 1.08 (NDCG) percentage points over the models without sentiment analysis and genres. Regarding the MARD dataset with β = 0.7 and L-CNN sentiment and genres-based model with SVD++ algorithms, the increase was 1.64 (MMR), 1.35 (MAP), and 0.31 (NDCG) percentage points over the approaches without sentiment and genre. If we consider all the results as a whole, we can conclude that, in general, the combination of the SVD++ method with the proposed model based on sentiment and genre is the one that provides the highest values of the three metrics: MMR, MAP and NDCG.

Regarding the evaluation of rating prediction, the results in Tables 5–8 show that RSME, MAE, and NMAE given by the approach that combines CF with sentiment analysis and genres are better than the error rates given by traditional CF methods without sentiment and genre on all algorithms. We found that the best results of the proposal are obtained with β = 0.7 on the MARD dataset and with β = 0.3 on the Amazon Movie database.

Figures 8–13 illustrate the comparison of the sentiment-based methods and genres with the L-CNN and C-LCTM with non-sentiment-based and genre methods with MARD and Amazon Movie datasets. We found that C-LSTM and L-CNN provide similar results. In addition, the sentiment-based and genre approach provides better results on Amazon Movie dataset.

Three algorithms (SVD, NMF, and SVD++) were tested in two ways, with explicit ratings only, and combining explicit ratings with sentiment extracted from reviews and genre embedding. As we mentioned, the genres attribute is preprocessed with advanced natural language processing techniques. Thus, our method is generalized to future data, such as other attributes of items, especially social tags, or other data generated by users. In most cases, the combined approach where two sentiment classification models (C-LSTM and L-CNN) are applied on music and movie review datasets gave better results than baselines tested. However, the improvement for top n recommendation is not as significant as that achieved for the rating prediction.

5. Conclusions

In this paper, we have proposed the use of sentiment analysis and genre embedding in streaming-service recommender systems. The approach is based on hybrid deep learning models, genre embedding, and user-based collaborative filtering methods. We conducted experiments with music and movie datasets containing information about item reviews and genres. Based on such experiments, we demonstrated the utility and applicability of our approach in producing personalized recommendations on online social networks. The improvements come from using sentiment analysis and item genres in a complementary way to the rating data to establish the similarity between users. Therefore, the recommendations provided by this approach are based on the affinity between users, both in terms of their preferences for items and in terms of the information underlying the reviews/genres they assign to them. In this way, the reliability of the recommendations is increased.

As future work, we plan to explore other application domains to ensure that the proposed architecture can efficiently solve similar problems. We also plan to address aspect sentiment analysis to gain deeper insight into user sentiments by associating them with specific features or topics, using the graph convolutional networks technique to improve this aspect. This technique seeks to predict the sentiment polarity of a sentence toward a specific aspect. Moreover, we plan to consider the aspect terms and the semantic and syntactic information by modeling their interaction to improve performance.

Author Contributions: Conceptualization, C.N.D. and M.N.M.-G.; methodology, M.N.M.-G. and F.D.l.P.; software, C.N.D.; validation, M.N.M.-G. and F.D.l.P.; formal analysis, C.N.D. and M.N.M.-G.; investigation, C.N.D.; data curation, C.N.D.; writing—original draft preparation, C.N.D.; writing—review and editing, M.N.M.-G. and F.D.l.P.; visualization, C.N.D.; supervision, M.N.M.-G. and F.D.l.P.; project administration, M.N.M.-G.; funding acquisition, M.N.M.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Junta de Castilla y León, Spain, grant number SA064G19.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Subramaniyaswamy, V.; Logesh, R.; Chandrashekhar, M.; Challa, A.; Vijayakumar, V. A personalised movie recommendation system based on collaborative filtering. *Int. J. High Perform. Comput.* **2017**, *10*, 54–63. [CrossRef]
2. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
3. Sánchez-Moreno, D.; González, A.B.G.; Vicente, M.D.M.; Batista, V.F.L.; García, M.N.M. A collaborative filtering method for music recommendation using playing coefficients for artists and users. *Expert Syst. Appl.* **2016**, *66*, 234–244. [CrossRef]
4. Sánchez-Moreno, D.; Zheng, Y.; Moreno-García, M.N. Time-aware music recommender systems: Modeling the evolution of implicit user preferences and user listening habits in a collaborative filtering approach. *Appl. Sci.* **2020**, *10*, 5324. [CrossRef]

5. Zhao, S.; King, I.; Lyu, M.R. Aggregated temporal tensor factorization model for point-of-interest recommendation. *Neural Process. Lett.* **2018**, *47*, 975–992. [CrossRef]
6. Sánchez-Moreno, D.; López Batista, V.F.; Muñoz Vicente, M.D.; Gil González, A.B.; Moreno-García, M.N. A session-based song recommendation approach involving user characterization along the play power-law distribution. *Complexity* **2020**, *2020*, 7309453. [CrossRef]
7. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [CrossRef]
8. Kastrati, Z.; Ahmedi, L.; Kurti, A.; Kadriu, F.; Murtezaj, D.; Gashi, F. A Deep Learning Sentiment Analyser for Social Media Comments in Low-Resource Languages. *Electronics* **2021**, *10*, 1133. [CrossRef]
9. Dang, C.N.; Moreno-García, M.N.; De la Prieta, F. Hybrid Deep Learning Models for Sentiment Analysis. *Complexity* **2021**, *2021*, 9986920. [CrossRef]
10. Kastrati, Z.; Dalipi, F.; Imran, A.S.; Pireva Nuci, K.; Wani, M.A. Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study. *Appl. Sci.* **2021**, *11*, 3986. [CrossRef]
11. Dang, C.N.; Moreno-García, M.N.; Prieta, F.D.L. An Approach to Integrating Sentiment Analysis into Recommender Systems. *Sensors* **2021**, *21*, 5666. [CrossRef]
12. Kumar, S.; De, K.; Roy, P.P. Movie recommendation system using sentiment analysis from microblogging data. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 915–923. [CrossRef]
13. Sánchez-Moreno, D.; Moreno-García, M.N.; Mobasher, B.; Sonboli, N.; Burke, R. Using Social Tag Embedding in a Collaborative Filtering Approach for Recommender Systems. In Proceedings of the 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Melbourne, Australia, 14–17 December 2020.
14. Rosa, R.L.; Rodriguez, D.Z.; Bressan, G. Music recommendation system based on user's sentiments extracted from social networks. *IEEE Trans. Consum. Electron.* **2015**, *61*, 359–367. [CrossRef]
15. Osman, N.A.; Noah, S.A.M. Sentiment-based model for recommender systems. In Proceedings of the 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), Kota Kinabalu, Malaysia, 26–28 March 2018; pp. 1–6.
16. Oramas, S.; Nieto, O.; Sordo, M.; Serra, X. A deep multimodal approach for cold-start music recommendation. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, Como, Italy, 27 August 2017; pp. 32–37.
17. Advani, G.; Soni, N. A Novel Way for Personalized Music Recommendation Using Social Media Tags. *IJSRD-Int. J. Sci. Res. Dev.* **2015**, *2*, 404–410.
18. Eck, D.; Lamere, P.; Bertin-Mahieux, T.; Green, S. Automatic generation of social tags for music recommendation. *Adv. Neural Inf. Process. Syst.* **2007**, *20*, 385–392.
19. Su, J.-H.; Chang, W.-Y.; Tseng, V.S. Personalized music recommendation by mining social media tags. *Procedia Comput. Sci.* **2013**, *22*, 303–312. [CrossRef]
20. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech* **2002**, *10*, 293–302. [CrossRef]
21. Lu, C.-C.; Tseng, V.S. A novel method for personalized music recommendation. *Expert Syst. Appl.* **2009**, *36*, 10035–10044. [CrossRef]
22. Jakubik, J.; Kwaśnicka, H. Similarity-based summarization of music files for Support vector machines. *Complexity* **2018**, *2018*, 1935938. [CrossRef]
23. Last.fm. Available online: https://www.last.fm/home (accessed on 24 July 2021).
24. MusicBrainz—The Open Music Encyclopedia. Available online: https://musicbrainz.org/ (accessed on 24 July 2021).
25. Reddy, S.; Nalluri, S.; Kunisetti, S.; Ashok, S.; Venkatesh, B. Content-based movie recommendation system using genre correlation. In *Smart Intelligent Computing and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 391–397.
26. Gunawan, A.A.; Suhartono, D. Music recommender system based on genre using convolutional recurrent neural networks. *Procedia Comput. Sci.* **2019**, *157*, 99–109.
27. Hong, J.; Deng, H.; Yan, Q. Tag-based artist similarity and genre classification. In Proceedings of the 2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, Wuhan, China, 21–22 December 2008; pp. 628–631.
28. MARD: Multimodal Album Reviews Dataset. Available online: https://www.upf.edu/web/mtg/mard (accessed on 24 July 2021).
29. McAuley, J.J.; Leskovec, J. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In Proceedings of the 22nd international conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 897–908.
30. Amazon Movie Database. Available online: https://www.amazon.com/dp/ (accessed on 27 August 2021).
31. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
32. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
33. Hochreiter, S.; Schmidhuber, J. LSTM can solve hard long time lag problems. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 2–5 December 1996; pp. 473–479.
34. Sallam, R.M.; Hussein, M.; Mousa, H.M. An Enhanced Collaborative Filtering-based Approach for Recommender Systems. *Int. J. Comput. Appl.* **2020**, *975*, 8887. [CrossRef]

35. Lara-Cabrera, R.; González-Prieto, Á.; Ortega, F. Deep matrix factorization approach for collaborative filtering recommender systems. *Appl. Sci.* **2020**, *10*, 4926. [CrossRef]
36. Xian, Z.; Li, Q.; Li, G.; Li, L. New collaborative filtering algorithms based on SVD++ and differential privacy. *Math. Probl. Eng.* **2017**, *2017*, 1975719. [CrossRef]
37. Salton, G.; McGill, M.J. *Introduction to Modern Information Retrieval*; Mcgraw-Hill: New York, NY, USA, 1983.
38. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
39. Making the Most of Your Colab Subscription. Available online: https://colab.research.google.com/notebooks/pro.ipynb (accessed on 22 January 2021).
40. Keras: The Python Deep Learning API. Available online: https://keras.io/ (accessed on 10 December 2020).
41. Pytorch. Available online: https://pytorch.org/ (accessed on 24 September 2021).
42. Surprise—A Python Scikit for Recommender Systems. Available online: http://surpriselib.com/ (accessed on 24 September 2021).
43. Chen, R.; Hua, Q.; Chang, Y.-S.; Wang, B.; Zhang, L.; Kong, X. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access* **2018**, *6*, 64301–64320. [CrossRef]