

Article

A Hybrid Recommendation System of Upcoming Movies Using Sentiment Analysis of YouTube Trailer Reviews

Sandipan Sahu ¹, Raghvendra Kumar ¹, Pathan MohdShafi ², Jana Shafi ³, SeongKi Kim ^{4,*}
and Muhammad Fazal Ijaz ^{5,*}

¹ Department of Computer Science and Engineering, GIET University, Rayagada 765022, India; sandipan.sahu@giet.edu (S.S.); raghvendra@giet.edu (R.K.)

² Department of Computer Science and Engineering, MITADT University, Loni Kalbhor 412201, India; shafi.pathan@mituniversity.edu.in

³ Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdul Aziz University, Wadi Ad Dawasir 11991, Saudi Arabia; j.jana@psau.edu.sa

⁴ National Centre of Excellence in Software, Sangmyung University, Seoul 03016, Korea

⁵ Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea

* Correspondence: skkim9226@smu.ac.kr (S.K.); fazal@sejong.ac.kr (M.F.I.)

Abstract: Movies are one of the integral components of our everyday entertainment. In today's world, people prefer to watch movies on their personal devices. Many movies are available on all popular Over the Top (OTT) platforms. Multiple new movies are released onto these platforms every day. The recommendation system is beneficial for guiding the user to a choice from among the overloaded contents. Most of the research on these recommendation systems has been conducted based on existing movies. We need a recommendation system for forthcoming movies in order to help viewers make a personalized decision regarding which upcoming new movies to watch. In this article, we have proposed a framework combining sentiment analysis and a hybrid recommendation system for recommending movies that are not yet released, but the trailer has been released. In the first module, we extracted comments about the movie trailer from the official YouTube channel for Netflix, computed the overall sentiment, and predicted the rating of the upcoming movies. Next, in the second module, our proposed hybrid recommendation system produced a list of preferred upcoming movies for individual users. In the third module, we finally were able to offer recommendations regarding potentially popular forthcoming movies to the user, according to their personal preferences. This method fuses the predicted rating and preferred list of upcoming movies from modules one and two. This study used publicly available data from The Movie Database (TMDb). We also created a dataset of new movies by randomly selecting a list of one hundred movies released between 2020 and 2021 on Netflix. Our experimental results established that the predicted rating of unreleased movies had the lowest error. Additionally, we showed that the proposed hybrid recommendation system recommends movies according to the user's preferences and potentially promising forthcoming movies.

Keywords: OTT platform; recommendation system; sentiment analysis; hybrid recommendation system; predicted rating

MSC: 94A15



Citation: Sahu, S.; Kumar, R.; MohdShafi, P.; Shafi, J.; Kim, S.; Ijaz, M.F. A Hybrid Recommendation System of Upcoming Movies Using Sentiment Analysis of YouTube Trailer Reviews. *Mathematics* **2022**, *10*, 1568. <https://doi.org/10.3390/math10091568>

Academic Editor: Seifedine Kadry

Received: 8 April 2022

Accepted: 3 May 2022

Published: 6 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In today's internet-connected world, lots of information is available online. Due to this overpressure of digital content, sometimes users are becoming overwhelmed. A recommendation system helps users to cope with this information explosion. As a result, Amazon, Netflix, other e-commerce platforms, and movie OTT platforms all use a recommendation system.

Recently, movie recommendations have become very popular due to the personalized approach of the user. Currently, people are most comfortable watching movies on their personal devices; as a result, most of the OTT platforms, such as Netflix, Amazon, Disney+, etc., have become very popular [1–3]. Recommendation systems can be classified into two broader types: collaborative and content-based filtering methods. Content-based (CBF) works by analyzing and examining the properties of items to make predictions. Collaborative filtering (CF) finds the similarities of different users or items to make predictions about what a particular user may like. CF approaches work based on the wisdom of the viewer. The viewer selected the items they liked most based on their tests and preferences. According to the authors of [4], high sparsity and user–item interaction are significant factors of recommendation systems. The best performance is introduced when content-based filtering (CBF) and collaborative filtering (CF) are both triggered.

CBF and CF have encountered a limitation as a cold start problem [5,6] and sparse rating problem [7–9], as many users do not rate items. Therefore, cold start problems are also categorized into two types: the new user cold-start problem and the new item cold-start problem. On the other hand, CBF only suffers from new user problems where no information about their likes/dislikes is available. Moreover, due to sparse user–item rating matrices, nearest neighbors or nearest items can be challenging to identify [10].

To overcome these limitations, comments on different social media platforms, such as YouTube, Twitter, and Quora, have been used [1]. Based on the sentiments in the comments, the content has been analyzed. Most of this research has been performed using sentiment analysis of comments about released movies on social media, but there are still some challenges for upcoming movies. For example, rating information and liking information are not available for impending movies; minimal metadata is available. Due to the scarcity of information, it is challenging to build a recommendation system for a unreleased movie.

In this study, we have developed a recommendation system that can effectively recommend upcoming movies to a user. In addition, we have also proposed a framework to predict the rating of the forthcoming movies using a sentiment analysis of YouTube trailer reviews.

To solve the above problems, first, we randomly selected a list of one hundred movies released between 2020 and 2021 on Netflix; we then scraped the movie trailer comments from the official YouTube channel of Netflix. Then, using the sentiment analysis, we computed the overall sentiment of the movie from the trailer reviews and predicted the rating of the unreleased movie. We also used movie data and movie intrinsic features from the TMDb dataset and new movie sets. Then, by combining the previous movie data and forthcoming movie data, we built a hybrid recommender system to produce a list of preferred upcoming movies. We were then finally able to offer potentially popular forthcoming movies to the user, according to their preferences. Our method fuses the predicted ranking and preferred list of upcoming movies from modules one and two.

Our experimental results show that the predicted rating of unreleased movies has a minimum error. Additionally, we have demonstrated that the proposed hybrid recommendation system recommends movies according to the user’s choices and potentially promising impending movies.

The main contributions of this research are:

1. This research work is one of the first studies to build a framework combining sentiment analysis and a hybrid recommendation system for recommending movies that are not yet released, but where only the trailer has been released.
2. We also proposed a model that predicts the movie’s rating before the movie’s official release by analyzing the sentiment of comments from trailer videos on YouTube.
3. We have proposed a new way of calculating the comprehensive sentiment of a unreleased movie.
4. We also proposed a new framework of a hybrid recommendation system, which can recommend an upcoming new movie to a user based on their preferences.

5. We have proposed an idea to calculate the weighting of each movie feature in order to calculate the similarity between two movies.

The rest of this paper is arranged as follows. Section 2 summarizes related studies on recommendation systems and sentiment analysis. Section 3 outlines our proposed framework in detail and discusses dataset descriptions. In Section 4, the proposed model is described elaborately. Section 5 presents the experimental results and a comparative study of other statistical models is shown and explained. Finally, in Section 6 our research contributions and their limitations as well as further research directions are discussed.

2. Related Works

In this section, we have presented a detailed survey on related previous studies. Our work includes a recommendation system and sentiment analysis to recommend the preferred items. Recommendation systems are primarily divided into three types [11–15]: collaborative filtering (CF) [16–18], content-based filtering (CBF) [19,20], and hybrid filtering [21–23]. Sentiment analysis is the mining of textual content to extract the internal information in the content [24–26]. These approaches are discussed as follows.

2.1. Recommendation System

CF is a procedure that can refine things that a user might prefer based on responses by similar users. Cold start is one of the significant problems of CF. Yang et al. [27] proposed a system that uses user records to infer ratings. If more the pages are read by a user, then this implies increased interest in the document. This system is helpful in overcoming the problem of cold start but is domain specific. Optimizing the recommendation system is a multidimensional problem. Therefore, various optimization algorithms have been proposed by researchers. Some of these are mentioned as follows. Katarya et al. [28] proposed a recommender system formed on grey wolf optimization and fuzzy c-mean on Movie lens data to predict user ratings. Hsu et al. [29] proposed an artificial bee colony optimization technique to implement a personalized auxiliary material recommendation system. Bobadilla et al. [30] used genetic algorithms to optimize the recommendation system. Ujjin et al. [31] proposed a particle swarm optimization compared with the genetic algorithm. Katarya et al. [32] improved their collaborative movie recommendation system [28] by using an artificial bee colony and k-mean cluster (ABC km). Their proposed system alleviates the cold start problem. Zhang et al. [33] proposed a framework that clusters users based on the user's profile attributes in order to implement a personalized, collaborative movie recommendation system. This work handles the time complexity concern of CF. Lavanya et al. [34] used implicit feedback and predicted ratings to implement CF. This approach reduces the data sparsity problem. Chen et al. [35] proposed a framework of CF where they built two separate lists for each user, one with the movies they like and the other with their disliked movies.

Content-based filtering (CBF) [19,20,36,37] is a popular and extensively researched recommendation system model. Content-based recommender systems utilize metadata information of items or textual items [38]. Linked open data (LOD) initiation suggests new ideas to extend item information with outside knowledge sources [39,40]. Uluyagmur et al. [41] used a content-based movie recommender that proposed which users and movie features are used. The proposed movie rating system used a movie feature dataset. Content-based movie recommendation systems consider different movie attributes such as movie genre, names of the actors, names of the directors, and other attributes to build a recommender system. Reddy et al. [42] proposed a recommendation system where which movie genre users preferred to watch was used to build a recommender system using the Movie Lens dataset. Correlations between content and attributes are measured to determine the similarity between items. Son et al. [43] proposed a multi-attribute network to calculate correlations and recommend items to users. The similarity between directly or indirectly correlated items is calculated using network analysis. Ali et al. [44] proposed a hybrid model where genomic tags of the movie were used with CBF to recommend movies with

similar tastes. Belarbi et al.'s [45] proposed model reduces the computational complexity by using principal component analysis (PCA) and Pearson correlation procedures to reduce redundant tags and dispense a low variance. Elahi et al. [46] leveraged the gap between high-level and low-level features in their study. They have used low-level feature colors, motion exceeds, and lighting from a film to make a hybrid recommendation system. A new movie recommendation system was proposed by Deldjoo et al. [47]. They have addressed the cold start problem for a new item. They have offered audio and visual descriptions extracted from videos and developed a video genome. Breiffuss et al. [48] built a knowledge graph by using user sentiment from existing movies. Based on the knowledge graph, they then developed a recommendation system.

Current research has revealed that the hybrid strategy (HRS) [21–23,49,50] is more effective than conventional strategies. In addition, hybrid techniques attenuate the deficiency of individual strategies due to the blend of both recommendation strategies. Melville et al. [51] designed a hybrid recommender system that used advanced content-based features in a collaborative model. This strategy improved the prediction and the sparsity problem. Zhang et al. [52] proposed an HRS using user recommender interaction. Taking input from the user recommended N items, based on the user's choice, it continues to use the recommender system. Walek et al. [53] proposed an HRS for movie recommendations using a fuzzy expert system. The proposed model uses a user's preferred and unpreferred genres, while a fuzzy expert system resolves the definitive list of recommended movies. Bahl et al. [54] proposed a hybrid movie recommendation system model combining CBF with CF. The matrix factorization (SVD) technique was used to implement collaborative filtering. Kumar et al. [55] developed a hybrid recommendation system for movies; the proposed system uses CF and CBF and sentiment analysis of tweets. Here, sentiment analysis has been used to determine current trends and user choices. Duan et al. [56] have proposed a hybrid recommendation system to offer users services based on an explicit rating mechanism using latent semantics (LSIER). The system perceives feedback from the users, and the LSIER model generates a list of recommended services.

2.2. Sentiment Analysis

Sentiment analysis [24–26,57–59] is a procedure for computationally determining and classifying individuals' sentiments as described in reviews, comments, or surveys as negative, positive, or neutral. Previous research has preliminarily analyzed the user's textual comments and classified the reviews into negative or positive sentiments. Sun et al. [60] developed a social media recommendation system framework using sentiment analysis. CF usually suffers from data sparsity problems. This work uses the inferred sentiment from the user feedback to improve the performance of the recommendation system. Hui et al. [61] proposed a model that mines users' social media information from their microblog. Then, the authors analyze the preference information and find the similarity between TV shows and online movies. The presented approach alleviates the cold start problem of the recommendation system. Diao et al. [62] proposed a probabilistic model using CF and topic modeling. They have used IMDB user reviews to track user interest and the distribution of movies' content. Revealing the user's interests and information about the movie helps the model create recommendations appropriately. Wang et al. [63] presented a Twitter sentiment analysis model. The model has combined the textual information of Twitter messages and sentiment diffusion patterns to achieve a better performance on sentiment analysis. Social media data can be used to understand user's characteristics. Dang et al. [64] proposed a recommendation system model that combined sentiment analysis of a users' social media data with collaborative filtering to achieve better accuracy. Data augmentation has been used to solve data scarcity issues, which affect the performance of deep learning (DL) models. Xiang et al. [65] proposed a lexical substitution for data augmentation (PLSDA) based on part-of-speech (POS) strategies to improve the performance of ML algorithms in sentiment analysis [24,66]. Effective computing and sentiment analysis are emerging research fields that leverage information retrieval, human–computer interaction,

and multimodal signal processing to discover people's sentiments from online social data. Cambria et al. [67] proposed a neuro symbolic framework based on commonsense to overcome the limitations of AI modules in the circumstance of sentiment analysis. They have used reproducible and unsupervised sub symbolic strategies to efficiently convert natural language into a protolanguage that interprets induced sentiment. Most of the studies in sentiment analysis consider only positive and negative sentences. We typically ignore neutral and ambivalence sentences, but in some studies [68,69], neutral or ambivalence sentiments have been used to improve the model's performance. Sentiment analysis from a conversation [70] is much more complex than single-sentence sentiment analysis, mainly because it presents contextual information. Another fine-grained sentiment analysis is aspect-based. Here, sentiment polarity is determined by a given aspect. Graph convolution neural networks [71–73] have been used in aspect-based sentiment analysis.

In most studies, a recommendation system was used for already existing released movies. Recommendation systems for upcoming movies are uncommon. Our proposed model is a hybrid recommendation system for efficiently recommending upcoming movies. First, we have used sentiment analysis to predict the rating of unreleased movies. Then, based on that, we recommend potentially good movies according to the user's personal preferences.

3. Material and Methods

This study aimed to develop a recommendation system for upcoming new movies based the choices of an individual user with the help of a hybrid recommendation system and sentiment analysis. Specifically, we have used viewer comments from movie trailer videos on YouTube for sentiment analysis.

3.1. Proposed Framework

Our proposed system has three significant models, which are described below. Figure 1 shows the proposed framework.

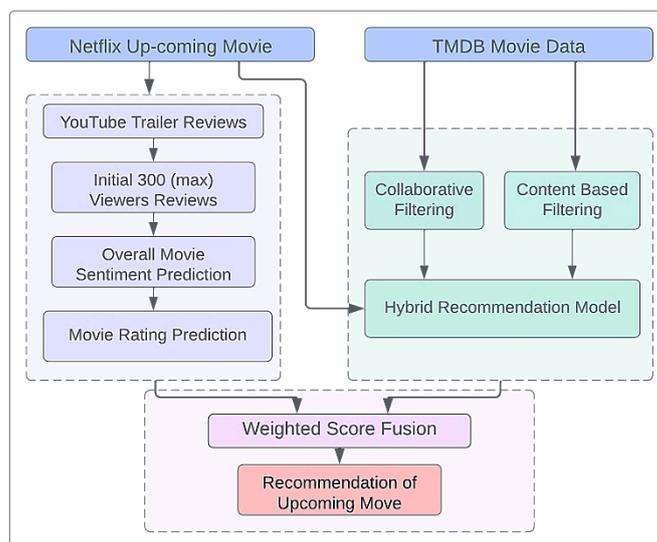


Figure 1. Framework workflow.

1. Movie trailers are usually released well before the release of the actual movie. Forthcoming movie trailers are generally available on YouTube. Viewers share their views about the trailer and express their thoughts regarding the unreleased movie by posting comments on the YouTube trailer video. In our proposed work, module one extracts the movie trailer comments from the official YouTube channel of Netflix. We then compute the overall sentiment and also predict the rating of the forthcoming movie.

2. In module 2, our objective is to produce a list of unreleased movies according to the preferences or likes of the individual user. Here, we have used the TMDb dataset, which contains movie metadata and user rating data. Firstly, we compute user preferences using the TMDb user rating dataset. Next, with the help of intrinsic movie data, we have discovered similar movies from the upcoming movie dataset that align with the user's taste. This module combines the previous and upcoming movies data and builds a hybrid recommendation system in order to produce a list of preferred upcoming movies.
3. The first module assesses the popularity of the forthcoming movies by predicting the rating of each new movie. The second module presents a list of new movies closely similar to the user's existing preferred movies. In the third module, we fuse the predicted ranking and preferred list of forthcoming movies from modules one and two. Finally, we are able to offer potentially popular unreleased movies to the user, according to their preference.

3.2. Dataset Description

The proposed method uses two separate datasets. Firstly, we have created a dataset of 100 movies representing a set of upcoming movies. This dataset consists of YouTube trailer reviews and some basic intrinsic features of each movie. Secondly, we used a publicly available TMDb dataset. The choice of this dataset is due to the presence of movie metadata and user rating data in a single dataset.

3.2.1. YouTube Trailer Review Dataset

In the proposed work, YouTube review data of movie trailers have been used. The objective is to analyze the sentiment of each review to predict the overall sentiment of an upcoming movie. User reviews of each movie trailer are scraped from the official YouTube channel of Netflix.

We randomly selected a list of one hundred movies released between 2020 and 2021 on Netflix. First, official trailers of each movie were identified on the Netflix YouTube channel. Then, we scraped each movie trailer's initial 300 (max) comments. In this study, comments were taken chronologically from the beginning to obtain initial responses from the viewers. Since movie trailers are published at least 1–2 months before the movie's official release, the initial comments we are using can be considered to have been given before the movie's release. Some examples of YouTube trailer review comments of the movie *The Starling* are shown in Table 1.

Table 1. Example of YouTube trailer review comments of the movie *The Starling*.

Reviewer Name	Comments	Time
Helen and Lolly	This film is going to break my heart. I can never have children. I have a long term illness. 😞	2021-09-06T16:00:12Z
Christina Watkins	Finally, a good Netflix movie trailer!!! Cannot wait to see Melissa play a more serious role 🌟❤️	2021-09-06T18:01:37Z
Samantha Richele	I'm already crying and it was just the trailer. 😞😭	2021-09-07T00:57:40Z

We have created a dataset of one hundred movies listed previously. For each movie, we have extracted the IMDb rating, movie genre, cast, director name, and keywords of the movie from the IMDb website (<https://www.imdb.com/> accessed on 2 May 2022). The

created data set was used as a set of new movies. Table 2 displays movie features and example values of a movie.

Table 2. Example of movie features extracted.

Features	Value
Original title	The Starling
IMDb Rating	6.3
Director	Theodore Melfi
Cast	Melissa McCarthy, Chris O’Dowd, Kevin Kline
Release Year	2021
Genres	Comedy, Drama
Keywords	woman adjusting life loss contends feisty bird garden husband who’s struggling find forward

3.2.2. TMDb Data Set

Several publicly available popular data sets are used for various recommendation systems. The different experiments were conducted using a movie dataset, such as the IMDb dataset, Movie lens 100 K, Movie lens 20 M, Netflix prize dataset, and TMDb data set.

This work uses the TMDb data set [74] to build a hybrid recommendation system. The dataset contains intrinsic movie features as well as user ratings. Movie features are used to obtain the content-based similarity between two movies, and user ratings are used to obtain the user preferences or choices. In our model, four movie features were used to compare the similarity between the two movies. The selected features are movie genre, director, cast (three primary cast members), and keywords. In addition, we also used IMDb movie ratings for collaborative filtering.

4. Experimental Methods

4.1. Analysis of Review Data

The scraping of YouTube comments was performed using Google script. For each movie trailer, 300 viewer comments from the beginning were scraped. All the selected movies were released on Netflix. Therefore, the trailer of each movie was released on the official YouTube channel of Netflix before the original movie’s release. As shown in Table 1, extracted review comments have tremendous noise, such as emojis, repetitive words, symbols, and other event data. Therefore, all the comments need to be cleaned using several preprocessing steps. Figure 2 shows the steps of data analysis.

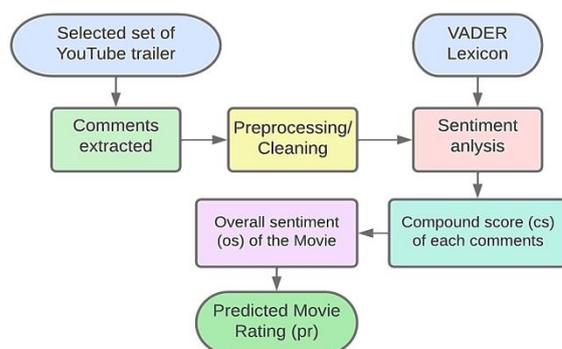


Figure 2. Framework of review data analysis.

4.1.1. Preprocessing of Review Data

To perform the sentiment analysis properly, data cleaning and preprocessing are necessary. In data cleaning, noise is removed by deleting the noise, and in the preprocessing step, the noise is cleaned by correcting the data. In the proposed work, data cleaning and data preprocessing have been performed to create purified review comments machine-understandable.

Multiple steps have been performed to clean the data, such as: removing emoji symbols, deleting words with only consonants, and deleting words with more than three repetitive and consecutive consonants and vowels. We also removed consecutive whitespaces. It is important to delete all the stop words. We deleted all unusable data, as shown in Table 3, which add no value in sentiment analysis. In the preprocessing part, lemmatization is one of the significant steps.

Table 3. Example of types of noise in the review comments.

Type of Noise	Example
Stop words	The, an, a, in, are, as, at, be
Words more than three same consecutive latter	2pacccccccccccc is better periodddddddddddd baby baby stole pacs style babyyyyyyyyyyyyyyyyyyyyyy
Weblink	May I know the background music name from https://www.youtube.com/watch?v=n4Uv5VHRDZg&t=0m33s >0:33 ❤️I just love it!
Special characters	#, @, !, \$, %, ... **
Emojis	😄 🤔 ❤️ 👍 ❤️

4.1.2. Sentiment Analysis

We have used VADER from the nltk sentiment analyzer to determine the sentiment’s polarity of each review comment. The VADER measures the intensity of each word of the review and determines the overall sentiment of the comments. For each comment, it produces values of four components of sentiment. The initial three components are positive, neutral, and negative. The value lies between [0, 1]. The fourth component is the compound score—the normalized value of positive, negative, and neutral scores. The domain value of the compounds score is between [−1, 1]. Where +1 represents extreme positive sentiment, and −1 represents extreme negative sentiment. Compounds with a score greater than zero (>0) classify the comment as a positive sentiment. Similarly, a lower compound score of less than zero (<0) is a comment with negative sentiment. Table 4 shows the values of four parameters for some example comments.

Table 4. Values of four parameters of some example comments.

Cleaned Comments	Positive	Neutral	Negative	Compound Score
looks really good hope movie trailerjitniacchi	0.515	0.485	0	0.7485
waste your time this boring chaotic with a stupid ending	0	0.294	0.706	−0.9006
basically, Netflix does not want people to sleep alwaysbinge-watching	0.14	0.86	0	0.0772

Overall Sentiment Score (os): In order to determine the overall sentiment of the movie from the individual comments, we have used the compound score (cs) of each comment. If

the compound score of a review is zero ($cs = 0$), this means the review holds a completely neutral sentiment. Neutral sentiment adds nothing to the positivity or negativity of the movie. Therefore, we dropped all the reviews with $cs = 0$. Next, we calculated the overall sentiment $os(m_i)$ of a movie m_i by taking the average of all the non-zero compound scores (cs). Equation (1) shows how to calculate $os(m_i)$.

$$os(m_i) = 1/k \sum_{j=1}^n cs(r_j^i) \tag{1}$$

In the equation, k is the total number of reviews (r) of the movie m_i for which $cs(r_j^i) \neq 0$. The overall score ranges between $[-1, 1]$.

Predicted Rating (pr): In this study, we calculated each movie’s predicted rating $pr(m_i)$ from the overall sentiment $os(m_i)$ that was computed earlier. The Overall sentiment is scaled in the range $[1, 10]$ using Equation (2). Table 5 shows some examples of the overall score $os(m_i)$ and the predicted rating $pr(m_i)$.

$$pr(m_i) = \{1 + os(m_i)\} * 5 \tag{2}$$

Table 5. Examples of some overall scores, predicted ratings, and IMDb ratings.

Movie Name	Overall Sentiment	Predicted Rating	IMDb Rating
<i>The Starling</i>	0.2359	6.2	6.3
<i>AjeebDaastaans</i>	0.3774	6.9	6.7
<i>Sentinelle</i>	0.0838	5.4	4.7
<i>Dance Dreams: Hot Chocolate</i>			
<i>Nutcracker</i>	0.5858	7.9	7.1

4.2. Hybrid Recommendation System

In the proposed work, a forthcoming movie is recommended to the user considering their preferences. Therefore, we have designed a hybrid recommendation system (HRS) model that combines collaborative and content-based filtering. We have shown the basic framework of the proposed HRS in Figure 3. Initially, a user-preferred list is created. Next, we find the similarity between new movies and that preferred movie set. Finally, we fuse the similarity score, predicted rating, and the rating of preferred movies to make a final recommendation of upcoming movies (Algorithm 1).

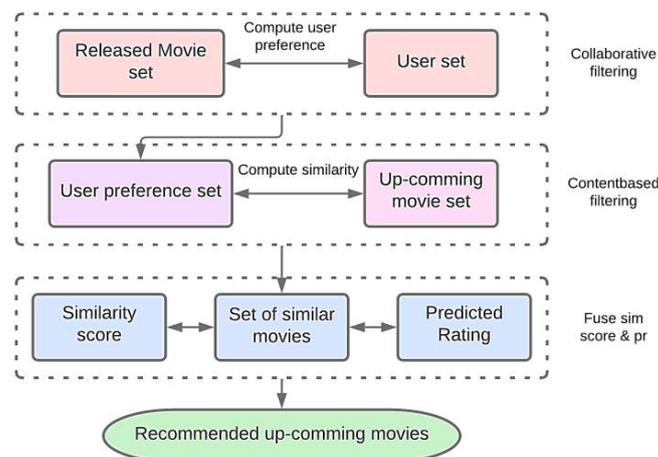


Figure 3. Framework of proposed hybrid recommendation system.

Algorithm 1: Hybrid Recommendation System for up-coming Movie Recommendations.

Input: Set of Users $U = \{u_1, u_2, \dots, u_n\}$
 Set of Movies $M = \{m_1, m_2, \dots, m_p\}$
 Set of New Movies $NM = \{nm_1, nm_2, \dots, nm_q\}$
 User Rating Matrix $R = U \times M$
 Movie Feature vector $F = \{f_1, f_2, \dots, f_m\}$
 Feature Weight vector $W = \{w_1, w_2, \dots, w_m\}$
 IMDb Rating of all M movies $R = \{r(m_1), r(m_2), \dots, r(m_p)\}$
 Predicted Rating of all NM new movies $PR = \{pr(nm_1), pr(nm_2), \dots, pr(nm_q)\}$
Output: Recommend R most promising upcoming movies RM_i according to the preference $u_i, \forall u_i \in U, RM^i = \{rm_j^i | j = 1, 2, \dots, r\}$.

1. **for** (Each user $u_i \in U$)
2. **Find** l preferred movies with the highest likes from $R = U \times M$
3. Preferred movie set $pm_i = \{m_1^i, m_2^i, \dots, m_l^i\}$ where $m_k^i \in M$ and $1 \leq k \leq l$
4. $PM = \{pm_1, pm_2, \dots, pm_l\}$
5. **for** (Each user $u_i \in U$)
6. **for** (Each movie $m_j^i \in pm_i$)
7. **for** (Each new movie $nm_k \in NM$)
8. $sim(m_j^i, nm_k) = \sum_{c=1}^{c=m} [w_c * sim_{fc}(m_j^i, nm_k)]$
9. $sim_j^i = \cup_{k=1}^{k=q} \{sim(m_j^i, nm_k)\}$ where $q =$ total no. of new movies
10. **Sort** (sim_j^i) [Most similar movie to less similar movie]
11. $NM_j^i = \{nm_{jk}^i | k = 1, 2, \dots, p\}$ consider only p most similar movies of m_j^i
12. $NM^i = \{NM_j^i | j = 1, 2, \dots, l\}$ Pool of $p * l$ no. of new movies for user u_i
13. **for** (Each movie $nm_{jk}^i \in NM^i$) [Make the final recommendation by fusing weighted score]
14. $cs(u_i, nm_{jk}^i) = sim(m_j^i, nm_{jk}^i) [r(m_j^i) + pr(nm_{jk}^i)]$ combined score of a new movie nm_{jk}^i
15. $CS^i = \cup_{j=1}^{j=l} \cup_{k=1}^{k=p} cs(u_i, nm_{jk}^i)$
16. **Sort** (CS^i) [Strongly recommended movies are with higher combined score cs]
17. $CS^i = \{nm_1^i, nm_2^i, \dots, nm_p^i\}$
18. $RM^i = \{rm_j^i | j = 1, 2, \dots, r\}$ considering only initial r movies to recommend
19. $RM = \{RM^i | i = 1, 2, \dots, n\}$
20. **Return** RM

In this study, we have used a user rating matrix $R = U \times M$, where $U = \{u_1, u_2, \dots, u_n\}$ set of users and $M = \{m_1, m_2, \dots, m_p\}$ is a set of existing movies. We then generated a preferred movie list $pm_i = \{m_1^i, m_2^i, \dots, m_l^i\}$ of l movies of each user u_i . Next, considering the preferred list pm_i and a user u_i , taking each movie from the list m_j^i we can find similar movies from the set of new movies $NM = \{nm_1, nm_2, \dots, nm_q\}$. For each preferred movie m_j^i we have considered p as the most similar new movies $NM_j^i = \{nm_{jk}^i | k = 1, 2, \dots, p\}$. In this manner, we created a pool of $p * l$ movies $NM^i = \{NM_j^i | j = 1, 2, \dots, l\}$. NM^i is a potential set of new movies for recommendation to the user u_i . Finally, we can make a final recommendation by fusing the weighted scores.

Movie similarity: The similarity between two movies is determined by using movie metadata. Let $F = (f_1, f_2, \dots, f_m)$ and $W = (w_1, w_2, \dots, w_m)$ represent the feature

vector of the movie and weight vector, respectively. The computed similarity between any two movies m_i and m_j concerning the feature f_k as:

$$sim_{f_k}(m_i, m_j) = \cos \text{ine_sim}(f_k^i, f_k^j) \tag{3}$$

The similarity between two movies is measured by combining the distance of m features. All the features do not have the same priority. We also have an m dimensional weight vector W . Equation (3) computed the cosine similarity of the feature values f_k between two movies m_i and m_j . We have used cosine similarity to find the distance. The overall similarity between the two movies m_i and m_j is computed using Equation (4).

$$sim(m_i, m_j) = \sum_{c=1}^{c=m} [w_c * sim_{f_c}(m_i, m_j)] \tag{4}$$

Features weight (W): To compute the weight vector W , we have used user preferences. Let $U = \{u_1, u_2, \dots, u_n\}$ where U is a set of users. $M_i = \{m_1^i, m_2^i, \dots, m_m^i\}$ is a set of m , the most preferred movies of the user u_i . In the same way $M = \{M_1, M_2, \dots, M_n\}$ is the collection of the preferred movie set of all users U . Considering each user u_i and the preferred set of movies M_i , we have discovered the importance of each movie feature. In this process, we first calculated the similarity between two movies m_i & $m_j \in M_i$ concerning feature f_k .

$$f_k(M_i) = \sum sim_{f_k}(m_i, m_j) \forall m_i, m_j \in M_i \& i \neq j \tag{5}$$

Equation (5) determines the total similarity value of the feature for the user u_i . Next, in Equation (6) we can determine the total similarity value of a feature f_k among all users U . This has been carried forward and generated $F(M)$ as the vector having a total similarity value of each feature f_k , considering all users U .

$$f_k(M) = \sum_{i=1}^n f_k(M_i) \tag{6}$$

$$F(M) = \{f_1(M), f_2(M), \dots, f_m(M)\} \tag{7}$$

In the vector $F(M)$, a higher value implies greater importance of the feature. Similarly, less essential features have a lower value. We have determined the feature weight vector W by normalizing the value of $F(M)$, and can thus generate $W = (w_1, w_2, \dots, w_m)$.

4.3. Weighted Score Fusion

In this work, our objective is to recommend new movies that should match three primary criteria: (1) Recommended new movie nm_{jk}^i must be a potentially good movie. (2) nm_{jk}^i has significant similarity with $m_j^i \in pm_i$. (3) Existing preferred movie m_j^i needs to be a good movie (with a higher IMDb rating). Using the sentiment analysis of the review comments, we have tried to figure out potential hit movies. We have assigned a predicted rating $pr(nm_i)$ to each of the new films nm_i . Using the hybrid recommendation system, we have been able to obtain the user's choice and find similar films from the set of new movies, thus creating a pool of $p * l$ movies NM^i for each user u_i . Each of the new movies $nm_{jk}^i \in NM^i$ is similar to the movie $m_j^i \in pm_i$ and the similarity is measured by $sim(m_j^i, nm_{jk}^i)$.

To make the final recommendation to the user u_i , we have used predicted rating $pr(nm)$ of the new movie $nm \in NM^i$ and the IMDb rating $r(m)$ of the corresponding similar movie $m \in pm_i$. We have then fused these two ratings using the similarity score $sim(m, nm)$. We have used Equation (8) to compute the combined score of the new movie nm .

$$cs(m, nm) = sim(m, nm)[r(m) + pr(nm)] \tag{8}$$

According to the combined r movies are selected $RM^i = \{rm_j^i | j = 1, 2, \dots, r\}$ to make a final recommendation to the user u_i .

5. Result and Discussion

5.1. New Movie Rating Prediction

We randomly selected one hundred movies released on Netflix in the proposed work from 2020 to 2021. We scraped comments from the official Netflix YouTube channel from each selected movie trailer. We analyzed each comment and predicted the movie’s overall rating, as discussed in Section 4.1. The movie’s rating was predicted by using the Vader lexicon and TextBlob lexicon models. Table 6 has taken some of the example movies and shows that the Vader model’s prediction is comparatively better than TextBlob. Vader predictions were much closer to the IMDB rating than the TextBlob model.

Table 6. Comparative analysis of predicted ratings of Vader and TextBlob with IMDB rating.

S. No.	Movie Name	IMDb Rating	Vader Rating	TextBlob Rating
1	<i>Caught by a Wave</i>	5.8	5.9	6.2
2	<i>The Starling</i>	6.4	6.2	6.2
3	<i>Squid Game</i>	8	7.1	6.3
4	<i>Dealer</i>	7.1	6.8	6.1
5	<i>Irul</i>	5.8	5.9	6.2
6	<i>The Midnight Sky</i>	5.6	5.4	5.1
7	<i>I Care a Lot</i>	6.3	5.8	5.4
8	<i>Ludo</i>	7.6	7.7	7.5
9	<i>Mank</i>	6.9	6.7	6.3
10	<i>The Devil All The Time</i>	7.1	6.6	6.6

From Figure 4, we can observe the performance of the Vader model and TextBlob. The presented performance results are computed considering all the selected movies. Vader performs significantly better in mean square error, R2-score, root mean square error, and mean absolute error.

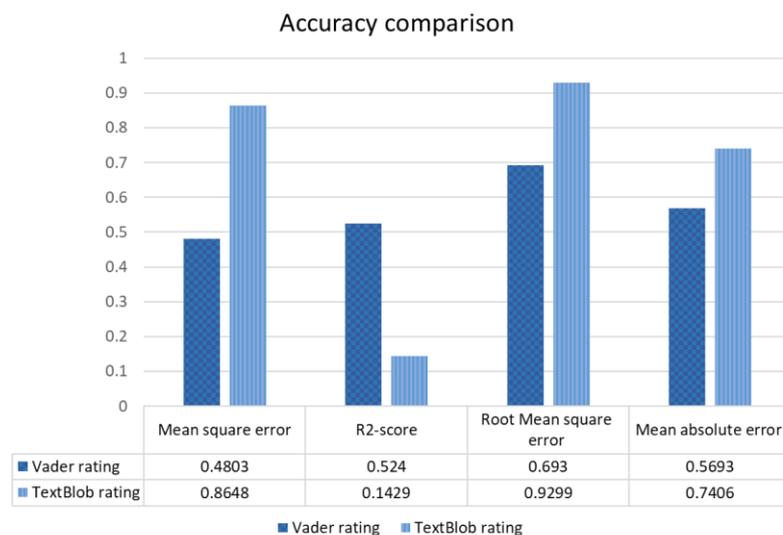


Figure 4. Comparative analysis of prediction error between Vader and TextBlob lexicons.

Figure 5 plotted the IMDB rating, Vader, and TextBlob predicted ratings for all 100 movies. The graph shows that both models are performing moderately well in general. However, the IMDB rating is drastically low for some specific movies, such as movie number 50. In those cases, Vader performs better than the TextBlob model. Again, for movies

number 55 and 60 the IMDB ratings are significantly higher than average. We can observe in the graph that the Vader model is also performing better here than the other model.

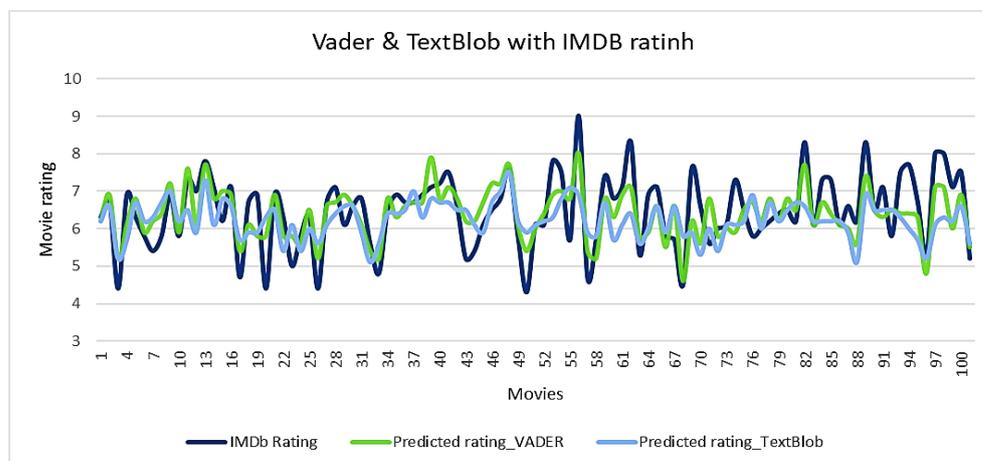


Figure 5. Comparison between Predicted rating by Vader and TextBlob and IMDb rating.

5.2. Movie Similarity

At first, we analyzed how content-based movie similarity performed. The similarity between two movies is computed using Equation (4) after calculating the similarity distance from each parameter. Table 7 shows similar movies to the movie *The Terminator*. Table 7 presents the 10 most similar movies to *The Terminator* and we also calculated the overall similarity from the selected movie to each of the similar movies. Additionally, we presented the genre and director of each of the listed movies. The similarity distance between two movies defines the similarity between them. According to the computation, *Terminator 2* is the most similar movie to *The Terminator*. The measured overall similarity is 0.7319 (73.19% similar); this is the minimum score among all the movies. The genres of the movie *The Terminator* are action, thriller, and science fiction. All the selected movies are also mostly grouped in the same genres. The director of the first four listed movies is “James Cameron”, who is also the director of *The Terminator*. Table 8 lists similar movies to the film *American Beauty*. The genre of the movie is drama. Here, all listed movies also have the same genre. The overall calculated similarity values are also shown in Table 8. According to the system, *Revolutionary Road* is the most similar movie to *American Beauty*, and the overall similarity distance is 0.4512 (i.e., 45.12% similar). Here, again, we can see that “Sam Mendes” is the director of the film *American Beauty* and the director of the top four similar movies listed in Table 8.

Table 7. Similar movies to the movie *The Terminator*.

Sl. No	Original Title	Genres	Director	Similarity
1	<i>Terminator2: Judgment Day</i>	Action, Thriller, Science Fiction	James Cameron	0.7319
2	<i>The Abyss</i>	Adventure, Action, Thriller, Science Fiction	James Cameron	0.5498
3	<i>Aliens</i>	Horror, Action, Thriller, Science Fiction	James Cameron	0.5498
4	<i>True Lies</i>	Action, Thriller	James Cameron	0.5375
5	<i>Terminator 3: Rise of the Machines</i>	Action, Thriller, Science Fiction	Jonathan Mostow	0.4534
6	<i>Terminator Genisys</i>	Science Fiction, Action, Thriller, Adventure	Alan Taylor	0.4139
7	<i>Avatar</i>	Action, Adventure, Fantasy, Science Fiction	James Cameron	0.3943
8	<i>Terminator Salvation</i>	Action, Science Fiction, Thriller	Mcg	0.3501
9	<i>The Running Man</i>	Action, Science Fiction	Paul Michael Glaser	0.3169
10	<i>Fortress</i>	Action, Thriller, Science Fiction	Stuart Gordon	0.3046

Table 8. Similar movies to the movie *American Beauty*.

Sl. No	Original Title	Genres	Director	Similarity
1	<i>Revolutionary Road</i>	Drama, Romance	Sam Mendes	0.4512
2	<i>Jarhead</i>	Drama, War	Sam Mendes	0.4268
3	<i>Road to Perdition</i>	Thriller, Crime, Drama	Sam Mendes	0.3943
4	<i>Away We Go</i>	Drama, Comedy, Romance	Sam Mendes	0.3943
5	<i>Regarding Henry</i>	Drama	Mike Nichols	0.3333
6	<i>The Wackness</i>	Drama	Jonathan Levine	0.313
7	<i>Albatross</i>	Drama	Niall MacCormick	0.2945
8	<i>The Cement Garden</i>	Drama	Andrew Birkin	0.2937
9	<i>Faces</i>	Drama	John Cassavetes	0.2886
10	<i>Liberty Heights</i>	Drama	Barry Levinson	0.2886

5.3. Hybrid Recommender System

In this section, we analyzed our proposed hybrid recommendation system (HRS). Our objective is to recommend potentially good upcoming movies according to each user’s previously liked movies. At first, we selected user 6 (u_6) randomly from the TMDB (user_rating) dataset. According to the dataset, u_6 had rated a total of 44 movies. Users can rate a movie between 5 and 0.5. Table 9 shows the rating distribution of the user u_6 .

Table 9. Distribution of ratings by user u_6 .

Rating	5	4.5	4	3.5	3	2.5	2	1.5	1	0.5
No. of movies	4	6	11	3	7	0	7	3	2	1

We considered only the top-rated (rating 5) movies of the user. User u_6 had only four movies with a rating of 5. Table 10 presents the list of the four movies that were used to create the user’s preferred movie list $pm_6 = \{m_1^6, m_2^6, m_3^6, m_4^6\}$ specifically for the user u_6 . All the selected movies have almost the same genres: adventure, action, and drama.

Table 10. Preferred movie set of a user.

Sl. No	Original Title	Genres	IMDB Rating
1	<i>The Lord of the Rings: The Return of the King</i>	Adventure, Fantasy, Action	9
2	<i>Taxi Driver</i>	Crime, Drama	8.3
3	<i>Lawrence of Arabia</i>	Adventure, Drama, History, War	8.3
4	<i>The Lord of the Rings: The Two Towers</i>	Adventure, Fantasy, Action	8.8

Next, we computed the similar movies from the set of new movies for each preferred movie. We have taken the five most similar movies of each preferred movie. In our new movie dataset we have 100 movies. From the set of those one hundred movies, we have created a pool of twenty movies NM^6 . Tables 11–14 present the five most similar movies of each preferred movie in Table 10.

Table 11. Similar movies to the movie *The Lord of the Rings: The Return of the King*.

	Original Title	Genres	Similarity	Predicted Rating VADER	Combined Score
1	<i>Jaguar</i>	Action, Adventure	0.2041	6.5	3.1639
2	<i>Project Power</i>	Action, Adventure	0.2041	6.2	3.1026
3	<i>Ganglands</i>	Crime, Action, Adventure	0.1666	6.3	2.55
4	<i>Thunder Force</i>	Action, Adventure, Comedy	0.1666	5.2	2.3666
5	<i>Dealer</i>	Crime, Action, Adventure	0.1443	6.8	2.2805

Table 12. Similar movies to the movie *Taxi Driver*.

Sl. No	Original Title	Genres	Similarity	Predicted Rating VADER	Combined Score
1	<i>The Trial of the Chicago 7</i>	Drama	0.1767	6.9	2.6870
2	<i>The White Tiger</i>	Drama	0.1767	6.7	2.6516
3	<i>All Together Now</i>	Drama	0.1767	6.5	2.6162
4	<i>Two Distant Strangers</i>	Drama	0.1767	6.1	2.5455
5	<i>Rogue City</i>	Action, Crime, Drama	0.1767	6.1	2.5455

Table 13. Similar movies to the movie *Lawrence of Arabia*.

Sl. No	Original Title	Genres	Similarity	Predicted Rating VADER	Combined Score
1	<i>Mosul</i>	Action, Adventure, Drama	0.1625	6.8	2.4537
2	<i>The Trial of the Chicago 7</i>	Drama	0.125	6.9	1.9
3	<i>The White Tiger</i>	Drama	0.125	6.7	1.875
4	<i>All Together Now</i>	Drama	0.125	6.5	1.85
5	<i>Two Distant Strangers</i>	Drama	0.125	6.1	1.8

Table 14. Similar movies to the movie *The Lord of the Rings: The Two Towers*.

Sl. No	Original Title	Genres	Similarity	Predicted Rating VADER	Combined Score
1	<i>Jaguar</i>	Action, Adventure	0.2041	6.5	3.1230
2	<i>Project Power</i>	Action, Adventure	0.2041	6.2	3.0618
3	<i>Ganglands</i>	Crime, Action, Adventure	0.1666	6.3	2.5166
4	<i>Thunder Force</i>	Action, Adventure, Comedy	0.1666	5.2	2.3333
5	<i>Dealer</i>	Crime, Action, Adventure	0.1443	6.8	2.2516

Table 11 presents the five similar movies to the film *The Lord of the Rings: The Return of the King*. The genres of this movie are adventure, fantasy, and action. All the selected movies are also within the same genres. The most similar movies are *Jaguar* and *Project Power*, and the similarity score is 0.2041. In Table 11, we have also shown the predicted ratings calculated by VADER of each new movie $pr(nm)$. We also presented a combined score $cs(m, nm)$ of each prereleased movie.

Table 12 presents five similar movies to the film *Taxi Driver*. The genres of the movie are crime and drama. All the selected movies also within the same genres. Here, the similarity scores are identical for all the selected movies, but the combined scores (cs) are different. The movie *The Trial of the Chicago 7* has the highest cs because it has the highest predicted rating (pr). Movie numbers 4 and 5 have the lowest cs because they have lowest pr .

Table 13 presents the five most similar movies to *Lawrence of Arabia*. The genres of the movie are adventure, drama, history, and war. The most similar movie has two genres in common with *Lawrence of Arabia*: adventure and drama. The other four movies only have one common genre: drama. The movie *Two Distant Strangers* has the lowest pr score; therefore, it has the lowest cs .

Table 14 presents five similar movies to *The Lord of the Rings: The Two Towers*. The movies *The Lord of the Rings: The Return of the King* and *The Lord of the Rings: The Two Towers* are highly similar with similarity score (0.8231). Therefore, they have generated the same list from the new movie set. However, the IMDb rating of *The Lord of the Rings: The Return of the King* (9) is slightly higher than *The Lord of the Rings: The Two Towers* (8.8), which affects the combined scores as shown in Tables 11 and 14. For example, the movie *Jaguar* has a cs in Table 11 of 3.1639 and in Table 14 the same movie has a cs of 3.1230.

5.4. Combined Score (CS)

We have produced a collection of movies, NM^6 as a potential set of new movies for recommendation to the user u_6 . Next, we can make the final selection by computing

the combined score $cs(m, nm)$ using weighted score fusion (WSF), as shown in Equation (8). To implement a WSF, we need the predicted rating $pr(nm)$ of each new movie, the IMDB rating $r(m)$ of each preferred movie, and the similarity score $sim(m, nm)$. We have computed the compound score (cs) of each movie in NM^6 . Next, we sort the movies according to their compound score (cs). Finally, we have made a final selection. Table 15 presents ten movies, which are finally recommended to the user u_6 . The recommended movies satisfy the preference of the user u_6 and also are all potentially good movies taken from all the available movies. In Table 15, we can observe that all the selected movies have high similarity scores and predicted ratings among all the movies in NM^6 .

Table 15. Recommended movies for user u_6 .

Sl. No	Original Title	Genres	Similarity	Predicted Rating VADER	Combined Score
1	Jaguar	Action, Adventure	0.2041	6.5	3.1639
2	Project Power	Action, Adventure	0.2041	6.2	3.1026
3	The Trial of the Chicago 7	Drama	0.1767	6.9	2.6870
4	The White Tiger	Drama	0.1767	6.7	2.6516
5	All Together Now	Drama	0.1767	6.5	2.6162
6	Ganglands	Crime, Action, Adventure	0.1666	6.3	2.55
7	Two Distant Strangers	Drama	0.1767	6.1	2.5455
8	Rogue City	Action, Crime, Drama	0.1767	6.1	2.5455
9	Mosul	Action, Adventure, Drama	0.1625	6.8	2.4537
10	Dealer	Crime, Action, Adventure	0.1443	6.8	2.2516

Figure 6 plots the combined score (cs) and the similarity scores (ss). The figure shows that the combined score and similarity score are directly proportionate. Therefore, with the increasing value of the similarity score, cs also increase. However, again, the increment of the combined score is not the same for all movies, and it depends on the predicted rating of new movies and the IMDB rating of the corresponding old movie. In the figure, scattered vertical lines indicate the difference between $pr(nm)$ and $r(m)$, which affects the value of cs .

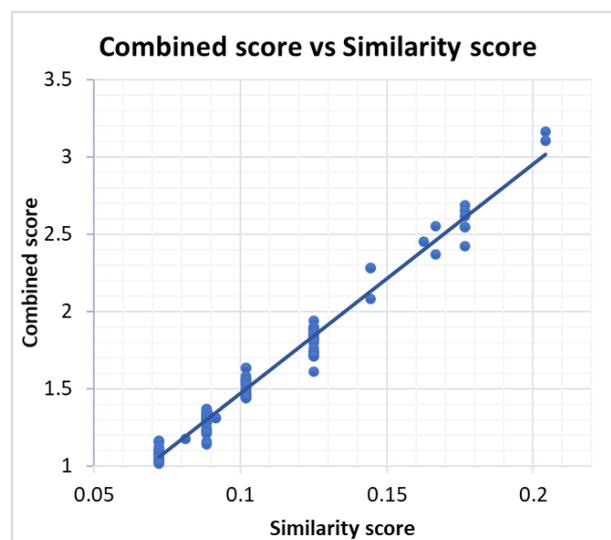


Figure 6. Relation between combined score and similarity score.

Figure 7 plots the normalized value of the combined score (cs) with the similarity score (ss) and the predicted rating (pr). We have scaled the values between 1 to 0. Figure 7 shows that the cs changes with the value of sc . Next, we plotted the decreasing similarity score value. Figure 7 shows that the combined score decreases with the decreasing value of sc .

One important thing that needs to be noticed is that cs is also influenced by the value of the predicted rating (pr). From movies number 3 to 8, the similarity scores remain constant, but the combined score constantly decreases with the decreasing value of the predicted rating. Again, for movie 9, cs increases as the value of pr increases. This shows that the predicted rating also influences the value of the combined score.

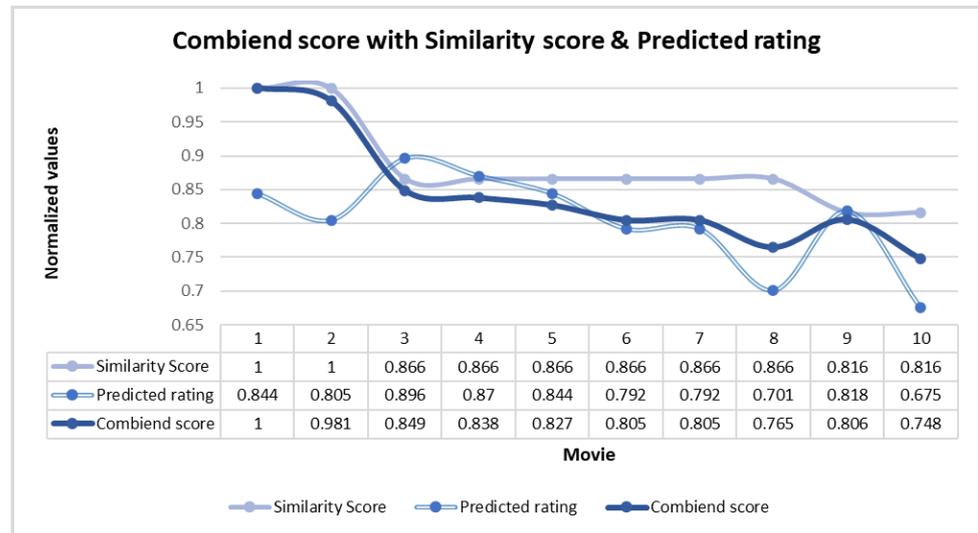


Figure 7. Relation between combined score with similarity score and predicted rating.

5.5. Qualitative Analysis

This section compares our proposed hybrid recommendation system with IMDb, TMDb recommendation system, and the model offered by Kumar et al. [55]. Although we have worked with only unreleased movies, we simulated our proposed hybrid recommendation system using the existing TMDb dataset to compare the current models qualitatively.

The Hollywood movie *Wonder Woman* was often recommended all the earlier systems. The movie’s genres are action, adventure, animation, and science fiction. The movie’s director is Lauren Montgomery and the IMDb rating is 7.2. Table 16 presents the recommended movies by our proposed HRS. Tables 17 and 18 show the recommended movies according to the IMDb and TMDb recommendation systems. Table 19 presents the recommended movies according to Kumar et al.’s model [55]. Movies in bold in Table 16 intersect with at least one of the other tables.

Table 16. Recommended movies for the movie *Wonder Woman* according to our proposed hybrid recommendation system.

Movie Name	Genres	IMDb Rating
<i>Justice League: Crisis on Two Earths</i>	Action, Adventure, Animation	7.1
<i>Batman: Year One</i>	Action, Adventure, Animation, Crime, Science Fiction	7.1
<i>Batman: Mask of the Phantasm</i>	Action, Adventure, Animation, Family	7.4
<i>Justice League: The Flashpoint Paradox</i>	Fantasy, Science Fiction, Animation, Action, Adventure	7.3
<i>Thor: Ragnarok</i>	Action, Adventure, Comedy	7.9
<i>Captain America: Civil War</i>	Adventure, Action, Science Fiction	7.8
<i>Batman: Under the Red Hood</i>	Action, Animation	7.6
<i>Batman: The Dark Knight Returns, Part 1</i>	Action, Animation	7.7
<i>Captain America: The Winter Soldier</i>	Action, Adventure, Science Fiction	7.6
<i>Batman v Superman: Dawn of Justice</i>	Action, Adventure, Fantasy	6.5

Table 17. Recommended movies for the movie *Wonder Woman* according to IMDb.

Movie Name	Genres	IMDb Rating
<i>Justice League</i>	Action, Adventure, Animation	7.1
<i>Batman v Superman: Dawn of Justice</i>	Action, Adventure, Fantasy	6.5
<i>Suicide Squad</i>	Action, Adventure, Crime, Fantasy, Science Fiction	5.9
<i>Thor: Ragnarok</i>	Action, Adventure, Comedy	7.9
<i>Spiderman: Homecoming</i>	Action, Adventure, Science Fiction	7.4
<i>Deadpool</i>	Action, Adventure, Comedy	7.4
<i>Logan</i>	Action, Drama, Science Fiction	7.6
<i>Captain America: Civil War</i>	Adventure, Action, Science Fiction	7.8
<i>Doctor Strange</i>	Action, Animation, Family, Fantasy, Science Fiction	6.6
<i>Guardians of the Galaxy Vol. 2</i>	Action, Adventure, Comedy, Science Fiction	7.6

Table 18. Recommended movies for the movie *Wonder Woman* according to TMDb.

Movie Name	Genres	IMDb Rating
<i>Guardians of the Galaxy Vol. 2</i>	Action, Adventure, Comedy, Science Fiction	7.6
<i>Spiderman: Homecoming</i>	Action, Adventure, Science Fiction	7.4
<i>Logan</i>	Action, Drama, Science Fiction	7.6
<i>Thor: Ragnarok</i>	Action, Adventure, Comedy	7.9
<i>Justice League</i>	Action, Adventure, Animation	7.1
<i>Pirates of the Caribbean: Dead Men Tell No Tales</i>	Action, Adventure, Fantasy	6.6
<i>Doctor Strange</i>	Action, Animation, Family, Fantasy, Science Fiction	6.6
<i>Baby Driver</i>	Action, Crime	7.2
<i>Kong: Skull Island</i>	Action, Adventure, Fantasy	6.2
<i>Life</i>	Comedy, Crime	6.4

Table 19. Recommended movies for the animated movie *Wonder Woman* according to the hybrid recommendation system proposed in [55].

Movie Name	Genres	IMDb Rating
<i>Batman v Superman: Dawn of Justice</i>	Action, Adventure, Fantasy	6.5
<i>Suicide Squad</i>	Action, Adventure, Crime, Fantasy, Science Fiction	5.9
<i>Thor: Ragnarok</i>	Action, Adventure, Comedy	7.9
<i>Justice League</i>	Action, Adventure, Animation	7.1
<i>Warcraft</i>	Action, Adventure, Fantasy	6.3
<i>Doctor Strange</i>	Action, Animation, Family, Fantasy, Science Fiction	6.6
<i>Guardians of the Galaxy Vol. 2</i>	Action, Adventure, Comedy, Science Fiction	7.6
<i>Kong: Skull Island</i>	Action, Adventure, Fantasy	6.2
<i>The LEGO Batman Movie</i>	Action, Animation, Comedy, Family, Fantasy	7.2
<i>Batman and Harley Quinn</i>	Animation, Action, Adventure	5.9

In Table 17, movies recommended by IMDb are all very similar to the movie *Wonder Woman*. We can observe that the movies in Tables 16 and 17 are closely related. However, movies such as *Suicide Squad* have a low IMDb rating, which is not present in Table 16. Similarly, in Tables 18 and 19, several movies do not have satisfactory IMDb ratings. Movies such as *Kong: Skull Island*, *Life*, *Warcraft*, and *Batman and Harley Quinn* are not high-quality movies according to IMDb ratings. All these films mentioned above are not included in Table 16. The average IMDb rating of all the movies listed in Table 16 is 7.4, which is highest among all these commendation systems. The average rating for the movies in Table 17 is 7.18, in Table 18 the average rating is 7.06, and Table 19 the average rating is 6.72—which is lowest among all the recommendation systems. *Wonder Woman* is an animated movie; most of the films (6 out of 10) recommended by our system in Table 16 are also animated movies. In Tables 17 and 18 only two animated movies are listed. Table 19 has four animated movies. Movies such as *Life* and *Baby Driver* present in Table 18 are not similar to the

animated movie *Wonder Woman*. Our proposed model recommended all the movies with high similarity and high IMDb ratings.

6. Conclusions and Future Works

The best method for information filtering is the recommendation system. Currently, this method is commonly used for handling vast amounts of data and meaningful mapping with the user. Our focus is on movie recommendations of forthcoming movies based on sentiment analysis of a user's social media. In our proposed model, we extracted the movie trailer comments from the official YouTube channel of Netflix. Then, we computed the overall sentiment and predicted an unreleased movie's rating. Secondly, we accepted movie data and a movie's intrinsic features from the TMDb dataset. Then, we combined the previous movie data and impending movie data by building a hybrid recommender system to produce a list of preferred upcoming movies. For sentiment analysis, we chose the Vader and TextBlob approaches where we predicted the Vader rating and TextBlob rating. When compared to the IMDb rating, an import factor was found. In most cases, Vader and TextBlob predictions showed accurate results, but in this experiment we found movie numbers 50, 55, and 60 showed some different results. This experiment only scraped YouTube comments from the official channel of Netflix in the English language. Furthermore, future studies could experiment with different social media platforms such as Twitter, etc., with other cross-lingual comments, could include more frequent access keywords such as "Lol" and "Omg", and could also consider emojis.

Author Contributions: Conceptualization, S.S., R.K. and P.M.; methodology, J.S., S.K. and M.F.I.; software, S.S., R.K. and P.M.; validation J.S., S.K. and M.F.I.; formal analysis, S.S., R.K., P.M., J.S., S.K. and M.F.I.; investigation and resources, P.M.; data duration, writing—review and editing, and visualization, S.S., R.K., P.M., J.S., S.K. and M.F.I.; supervision, S.K. and M.F.I.; project administration, S.K. and M.F.I.; funding acquisition, S.K. and M.F.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP (Institute of Information & communications Technology Planning & Evaluation) in 2022(2019-0-01880).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Jana Shafi would like to thank the Deanship of Scientific Research, Prince Sattam bin Abdul Aziz University, for supporting this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Carrer-Neto, W.; Hernández-Alcaraz, M.L.; Valencia-García, R.; García-Sánchez, F. Social knowledge-based recommender system. Application to the movies domain. *Expert Syst. Appl.* **2012**, *39*, 10990–11000. [[CrossRef](#)]
2. Winoto, P.; Tang, T.Y. The role of user mood in movie recommendations. *Expert Syst. Appl.* **2010**, *37*, 6086–6092. [[CrossRef](#)]
3. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 233–240.
4. Ma, H.; Yang, H.; Lyu, M.R.; King, I. Sorec: Social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 931–940.
5. Schein, A.I.; Popescul, A.; Ungar, L.H.; Pennock, D.M. Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11–15 August 2002; pp. 253–260.
6. Ma, H.; Zhou, T.C.; Lyu, M.R.; King, I. Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst. (TOIS)* **2011**, *29*, 1–23. [[CrossRef](#)]
7. Sarwar, B.M. *Sparsity, Scalability, and Distribution in Recommender Systems*; University of Minnesota: Minneapolis, MN, USA, 2001.
8. Huang, Z.; Chen, H.; Zeng, D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst. (TOIS)* **2004**, *22*, 116–142. [[CrossRef](#)]

9. Natarajan, S.; Vairavasundaram, S.; Natarajan, S.; Gandomi, A.H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Syst. Appl.* **2020**, *149*, 113248. [[CrossRef](#)]
10. Lu, Z.; Dou, Z.; Lian, J.; Xie, X.; Yang, Q. Content-based collaborative filtering for news topic recommendation. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
11. Sharma, L.; Anju, G. A survey of recommendation system: Research challenges. *Int. J. Eng. Trends Technol. (IJETT)* **2013**, *4*, 1989–1992.
12. Das, N.; Borra, S.; Dey, N.; Borah, S. Social networking in web based movie recommendation system. In *Social Networks Science: Design, Implementation, Security, and Challenges*; Springer: Cham, Switzerland, 2018; pp. 25–45.
13. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. *Egypt. Inform. J.* **2015**, *16*, 261–273. [[CrossRef](#)]
14. Shani, G.; Gunawardana, A. Evaluating recommendation systems. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 257–297.
15. Melville, P.; Sindhvani, V. Recommender systems. *Encycl. Mach. Learn.* **2010**, *1*, 829–838.
16. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [[CrossRef](#)]
17. Koren, Y.; Bell, R. Advances in collaborative filtering. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 77–118.
18. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**, *2009*, 421425. [[CrossRef](#)]
19. Lops, P.; Gemmis, M.D.; Semeraro, G. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 73–105.
20. Pazzani, M.J.; Billsus, D. Content-based recommendation systems. In *The Adaptive Web*; Springer: Heidelberg/Berlin, Germany, 2007; pp. 325–341.
21. Burke, R. Hybrid recommender systems: Survey and experiments. *User Modeling User-Adapt. Interact.* **2002**, *12*, 331–370. [[CrossRef](#)]
22. Burke, R. Hybrid web recommender systems. In *The Adaptive Web*; Springer: Heidelberg/Berlin, Germany, 2007; pp. 377–408.
23. Paradarami, T.K.; Bastian, N.D.; Wightman, J.L. A hybrid recommender system using artificial neural networks. *Expert Syst. Appl.* **2017**, *83*, 300–313. [[CrossRef](#)]
24. Cambria, E.; Das, D.; Bandyopadhyay, S.; Feraco, A. Affective computing and sentiment analysis. In *A Practical Guide to Sentiment Analysis*; Springer: Cham, Switzerland, 2017; pp. 1–10.
25. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [[CrossRef](#)]
26. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. *arXiv* **2002**, arXiv:cs/0205070.
27. Yang, C.; Wei, B.; Wu, J.; Zhang, Y.; Zhang, L. CARES: A ranking-oriented CADAL recommender system. In Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries, New York, NY, USA, 15–19 June 2009; pp. 203–212.
28. Katarya, R.; Verma, O.P. Recommender system with grey wolf optimizer and FCM. *Neural Comput. Appl.* **2018**, *30*, 1679–1687. [[CrossRef](#)]
29. Hsu, C.C.; Chen, H.C.; Huang, K.K.; Huang, Y.M. A personalized auxiliary material recommendation system based on learning style on Facebook applying an artificial bee colony algorithm. *Comput. Math. Appl.* **2012**, *64*, 1506–1513. [[CrossRef](#)]
30. Bobadilla, J.; Ortega, F.; Hernando, A.; Alcalá, J. Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowl.-Based Syst.* **2011**, *24*, 1310–1316. [[CrossRef](#)]
31. Ujjin, S.; Bentley, P.J. Particle swarm optimization recommender system. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS'03 (Cat. No. 03EX706), Indianapolis, IN, USA, 26 April 2003; pp. 124–131.
32. Katarya, R. Movie recommender system with metaheuristic artificial bee. *Neural Comput. Appl.* **2018**, *30*, 1983–1990. [[CrossRef](#)]
33. Zhang, J.; Wang, Y.; Yuan, Z.; Jin, Q. Personalized real-time movie recommendation system: Practical prototype and evaluation. *Tsinghua Sci. Technol.* **2019**, *25*, 180–191. [[CrossRef](#)]
34. Lavanya, R.; Bharathi, B. Movie Recommendation System to Solve Data Sparsity Using Collaborative Filtering Approach. *Trans. Asian Low-Resour. Lang. Inf. Processing* **2021**, *20*, 1–14. [[CrossRef](#)]
35. Chen, Y.L.; Yeh, Y.H.; Ma, M.R. A movie recommendation method based on users' positive and negative profiles. *Inf. Processing Manag.* **2021**, *58*, 102531. [[CrossRef](#)]
36. Philip, S.; Shola, P.; Ovyte, A. Application of content-based approach in research paper recommendation system for a digital library. *Int. J. Adv. Comput. Sci. Appl.* **2014**, *5*, 37–40. [[CrossRef](#)]
37. Viard, T.; Fournier-S'niehotta, R. Movie rating prediction using content-based and link stream features. *arXiv* **2018**, arXiv:1805.02893.
38. Belkin, N.J.; Croft, W.B. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM* **1992**, *35*, 29–38. [[CrossRef](#)]
39. Di Noia, T.; Mirizzi, R.; Ostuni, V.C.; Romito, D.; Zanker, M. Linked open data to support content-based recommender systems. In Proceedings of the 8th International Conference on Semantic Systems, Graz, Austria, 5–7 September 2012; pp. 1–8.

40. Musto, C.; Lops, P.; de Gemmis, M.; Semeraro, G. Semantics-aware recommender systems exploiting linked open data and graph-based features. *Knowl.-Based Syst.* **2017**, *136*, 1–14. [[CrossRef](#)]
41. Uluyagmur, M.; Cataltepe, Z.; Tayfur, E. Content-based movie recommendation using different feature sets. In Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 24–26 October 2012; Volume 1, pp. 17–24.
42. Reddy, S.R.S.; Nalluri, S.; Kuniseti, S.; Ashok, S.; Venkatesh, B. Content-based movie recommendation system using genre correlation. In *Smart Intelligent Computing and Applications*; Springer: Singapore, 2019; pp. 391–397.
43. Son, J.; Kim, S.B. Content-based filtering for recommendation systems using multiattribute networks. *Expert Syst. Appl.* **2017**, *89*, 404–412. [[CrossRef](#)]
44. Ali, S.M.; Nayak, G.K.; Lenka, R.K.; Barik, R.K. Movie recommendation system using genome tags and content-based filtering. In *Advances in Data and Information Sciences*; Springer: Singapore, 2018; pp. 85–94.
45. Belarbi, M.A.; Mahmoudi, S.; Belalem, G. PCA as dimensionality reduction for large-scale image retrieval systems. *Int. J. Ambient. Comput. Intell. (IJACI)* **2017**, *8*, 45–58. [[CrossRef](#)]
46. Elahi, M.; Deldjoo, Y.; Bakhshandegan Moghaddam, F.; Cella, L.; Cereda, S.; Cremonesi, P. Exploring the semantic gap for movie recommendations. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 326–330.
47. Deldjoo, Y.; Dacrema, M.F.; Constantin, M.G.; Eghbal-Zadeh, H.; Cereda, S.; Schedl, M.; Ionescu, B.; Cremonesi, P. Movie genome: Alleviating new item cold start in movie recommendation. *User Modeling User-Adapt. Interact.* **2019**, *29*, 291–343. [[CrossRef](#)]
48. Breiffuss, A.; Errou, K.; Kurteva, A.; Fensel, A. Representing emotions with knowledge graphs for movie recommendations. *Future Gener. Comput. Syst.* **2021**, *125*, 715–725. [[CrossRef](#)]
49. Aslanian, E.; Radmanesh, M.; Jalili, M. Hybrid recommender systems based on content feature relationship. *IEEE Trans. Ind. Inform.* **2016**. [[CrossRef](#)]
50. Porcel, C.; Tejada-Lorente, A.; Martínez, M.A.; Herrera-Viedma, E. A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Inf. Sci.* **2012**, *184*, 1–19. [[CrossRef](#)]
51. Melville, P.; Mooney, R.J.; Nagarajan, R. Content-boosted collaborative filtering for improved recommendations. In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002), Edmonton, Canada, 28 July–1 August 2002; Volume 23, pp. 187–192.
52. Zhang, H.R.; Min, F.; He, X.; Xu, Y.Y. A hybrid recommender system based on user-recommender interaction. *Math. Probl. Eng.* **2015**, *2015*, 145636. [[CrossRef](#)]
53. Walek, B.; Fojtik, V. A hybrid recommender system for recommending relevant movies using an expert system. *Expert Syst. Appl.* **2020**, *158*, 113452. [[CrossRef](#)]
54. Bahl, D.; Kain, V.; Sharma, A.; Sharma, M. A novel hybrid approach towards movie recommender systems. *J. Stat. Manag. Syst.* **2020**, *23*, 1049–1058. [[CrossRef](#)]
55. 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](#)]
56. Duan, L.; Gao, T.; Ni, W.; Wang, W. A hybrid intelligent service recommendation by latent semantics and explicit ratings. *Int. J. Intell. Syst.* **2021**, *36*, 7867–7894. [[CrossRef](#)]
57. Prabowo, R.; Thelwall, M. Sentiment analysis: A combined approach. *J. Informetr.* **2009**, *3*, 143–157. [[CrossRef](#)]
58. Ravi, K.; Ravi, V. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowl.-Based Syst.* **2015**, *89*, 14–46. [[CrossRef](#)]
59. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [[CrossRef](#)]
60. Sun, J.; Wang, G.; Cheng, X.; Fu, Y. Mining affective text to improve social media item recommendation. *Inf. Processing Manag.* **2015**, *51*, 444–457. [[CrossRef](#)]
61. Li, H.; Cui, J.; Shen, B.; Ma, J. An intelligent movie recommendation system through group-level sentiment analysis in microblogs. *Neurocomputing* **2016**, *210*, 164–173. [[CrossRef](#)]
62. Diao, Q.; Qiu, M.; Wu, C.Y.; Smola, A.J.; Jiang, J.; Wang, C. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 193–202.
63. Wang, L.; Niu, J.; Yu, S. SentiDiff: Combining textual information and sentiment diffusion patterns for Twitter sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 2026–2039. [[CrossRef](#)]
64. 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](#)] [[PubMed](#)]
65. Xiang, R.; Chersoni, E.; Lu, Q.; Huang, C.R.; Li, W.; Long, Y. Lexical data augmentation for sentiment analysis. *J. Assoc. Inf. Sci. Technol.* **2021**, *72*, 1432–1447. [[CrossRef](#)]
66. Cambria, E. Affective Computing and Sentiment Analysis. *IEEE Intell. Syst.* **2016**, *31*, 102–107. [[CrossRef](#)]
67. Cambria, E.; Liu, Q.; Decherchi, S.; Xing, F.; Kwok, K. SenticNet 7: A Commonsense-based Neurosymbolic AI Framework for Explainable Sentiment Analysis. LREC. 2022. Available online: <https://sentic.net/publications/#sentiment-analysis> (accessed on 2 May 2022).

68. Wang, Z.; Ho, S.B.; Cambria, E. Multi-level fine-scaled sentiment sensing with ambivalence handling. *Int. J. UncertainFuzziness Knowl.-Based Syst.* **2020**, *28*, 683–697. [[CrossRef](#)]
69. Valdivia, A.; Luzón, M.V.; Cambria, E.; Herrera, F. Consensus vote models for detecting and filtering neutrality in sentiment analysis. *Inf. Fusion* **2018**, *44*, 126–135. [[CrossRef](#)]
70. Li, W.; Shao, W.; Ji, S.; Cambria, E. BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis. *Neurocomputing* **2022**, *467*, 73–82. [[CrossRef](#)]
71. Liang, B.; Su, H.; Gui, L.; Cambria, E.; Xu, R. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowl.-Based Syst.* **2022**, *235*, 107643. [[CrossRef](#)]
72. Zhao, M.; Yang, J.; Zhang, J.; Wang, S. Aggregated graph convolutional networks for aspect-based sentiment classification. *Inf. Sci.* **2022**, *600*, 73–93. [[CrossRef](#)]
73. Dai, A.; Hu, X.; Nie, J.; Chen, J. Learning from word semantics to sentence syntax by graph convolutional networks for aspect-based sentiment analysis. *Int. J. Data Sci. Anal.* **2022**, 1–10. [[CrossRef](#)]
74. Available online: <https://www.kaggle.com/tmdb/tmdb-movie-metadata> (accessed on 2 May 2022).