





Review

# Recommender Systems Based on Collaborative Filtering Using Review Texts—A Survey

Mehdi Srifi <sup>1,\*</sup> , Ahmed Oussous <sup>2</sup> , Ayoub Ait Lahcen <sup>1,2</sup>  and Salma Mouline <sup>1</sup> 

<sup>1</sup> LRIT, Associated Unit to CNRST (URAC 29), Mohammed V University, Rabat 10090, Morocco; ayoub.aitlahcen@univ-ibntofail.ac.ma (A.A.L.); salma.mouline@um5.ac.ma (S.M.)

<sup>2</sup> LGS, National School of Applied Sciences (ENSA), Ibn Tofail University, Kenitra 14000, Morocco; ahmed.oussous@uit.ac.ma

\* Correspondence: mehdi\_srifi@um5.ac.ma

Received: 16 May 2020; Accepted: 9 June 2020; Published: 12 June 2020



**Abstract:** In e-commerce websites and related micro-blogs, users supply online reviews expressing their preferences regarding various items. Such reviews are typically in the textual comments form, and account for a valuable information source about user interests. Recently, several works have used review texts and their related rich information like review words, review topics and review sentiments, for improving the rating-based collaborative filtering recommender systems. These works vary from one another on how they exploit the review texts for deriving user interests. This paper provides a detailed survey of recent works that integrate review texts and also discusses how these review texts are exploited for addressing some main issues of standard collaborative filtering algorithms.

**Keywords:** recommender systems; collaborative filtering; user reviews; text mining; opinion mining; survey

## 1. Introduction

Nowadays, e-commerce websites have been flourishing quickly and permitting millions of items for selling [1]. The choice of an item from this large number of items makes necessary the use of a supplementary tool called recommender system [2,3]. The recommender system (RS) provides an alternative to discover items that users might not have found by themselves. It collects user's information concerning the items he/she prefers and then suggests those items [4].

One of the most widely used recommender systems rely on the Collaborative Filtering (CF) approach, which is utilized by various e-commerce companies [5], including Yelp (<https://www.yelp.com/>), Netflix (<https://www.netflix.com/>), eBay (<https://www.ebay.com/>), and Amazon (<https://www.amazon.com/>). The mainstream of CF techniques relies on the commonality between users. Analogous users or items are discovered by computing the similarities of the users' common ratings [4]. CF methods perform well when there is enough rating information [6]. Nevertheless, their effectiveness suffers when the rating sparsity issue occurs, for the reason that there are frequently a restricted common ratings' number between users [7]. Another limitation is that CF approaches do not catch the reason for ratings of the user, and consequently cannot precisely catch the preference of a target user [8]. To deal with these problems, several content-based methods have been developed to represent users and items by various kinds of data, including tags [9], items' descriptions [10], and social factors [11]. After all, these techniques are still deficient, particularly when the rating sparsity degree is major, or the target user has not much historical ratings [6]. With the current scenario of the Web, users have become more and more comfortable with expressing themselves and sharing their points of view concerning items on the e-platforms utilizing textual reviews [12]. As a result, user textual reviews have developed into an omnipresent portion of e-commerce nowadays. Forum websites, like TripAdvisor (<https://www.tripadvisor.com/>) and Yelp, and online retail websites such as

Amazon and Taobao (<https://www.taobao.com/>), are collecting huge amounts of online reviews [7]. Both companies and consumers benefit significantly from the valuable and rich knowledge contained in reviews [13]. Compared to rating information, textual reviews have more semantic information which provides a recommender system with more fine-grained, nuanced, and reliable user preference information [12]. Consequently, the system can construct a detailed preference representation for the user, that cannot be derived from global rating scores [6].

Recently, many efforts have been dedicated for capturing user interest information from review texts for rating prediction purposes [6]. Findings of these researches have proven review texts' positive impact on the performance of standard rating-based systems [6,12,14]. Therefore, this paper focuses on user review texts and surveys the recent studies that integrate the rich information contained in reviews in order to mitigate the main issues of the standard rating-based systems like sparsity and prediction accuracy problems.

The rest of this article is structured as follows. Section 2 reviews the traditional CF algorithms and their principal issues. Section 3 presents the main elements of user reviews. Section 4 summarizes the recent studies that integrate user reviews into CF systems. Section 5 discusses the practical benefits of these studies. Finally, Section 6 concludes this survey article.

## 2. Standard CF-Based Recommendation Techniques

A CF-based RS utilizes ratings for items provided by a group of users [15]. It suggests items that the target user has not yet regarded but will be appreciated [4]. Ratings are saved in an  $m \times n$  matrix, where  $m$  refers to the number of users and  $n$  represents the number of items (Table 1). The matrix rows store the ratings that users have scored towards items, and columns stock the ratings every item has been obtained [16]. A novel empty row is added to the matrix when a new user joins the system. Likewise, a novel empty column is added when a new item is putted in the catalog.

**Table 1.** An example of a rating matrix [15].

| User/Item | K-Pax | Life of Brian | Memento | Notorious |
|-----------|-------|---------------|---------|-----------|
| Alice     | 4     | 3             | 2       | 4         |
| Bob       | ∅     | 4             | 5       | 5         |
| Cindy     | 2     | 2             | 4       | ∅         |
| David     | 3     | ∅             | 5       | 2         |

A CF system generates recommendations based on the relationships and similarities between users or items [17]. These relations are inferred from the user-item interactions managed by the RS. This later infers the ratings of the target user for the items that have not been evaluated yet. After that, items are ranked according to the estimated rating scores, and then items with high-ranking are suggested to the targeted user [17].

### 2.1. Typical Algorithms of CF

CF is considered the widely studied and implemented approach in RS [4]. Existing CF can be classified into two principal categories of memory and model-based techniques [17,18]. In memory-based CF (also called Neighborhood-based), the ratings matrix saved in the system is straightly utilized to predict missing ratings for target items. Instead, model-based CF exploits the values of the matrix to build a model, which is then utilized to infer the pertinence of novel items for the target users [17].

#### 2.1.1. Memory-Based CF

Memory-based CF approach leverages on the similarities between users or items for inferring the user's probable preference in items which he has not evaluated previously. The memory-based CF method is subclustered into two main classes, namely, user-based and item-based methods [2]. The user-based CF predicts the unknown ratings of the user on the target items based on ratings of

similar users on given items [17]. Formally, the rating prediction of the user  $u$  to the item  $j$  is calculated as follows:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_u} sim(u,v) \times (r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u} |sim(u,v)|}, \tag{1}$$

where  $\bar{r}_u$  refers to the average rating of user  $u$ ,  $sim(u,v)$  is the similarity (for a predefined similarity metric) of the users  $u$  and  $v$ , and  $N_u$  represents a group of users similar to user  $u$  (neighbors) who rated item  $i$ .

The item-based CF relies on the similarities between items. It predicts the rating of the user for an item based on the user's ratings for similar items [17]. In these techniques, two items are similar if multiple users have evaluated these items similarly [4]. The rating prediction for item-based CF is formulated as follows:

$$\hat{r}_{u,j} = \frac{\sum_{k \in N_i} sim(j,k) \times r_{u,k}}{\sum_{k \in N_i} |sim(j,k)|}, \tag{2}$$

where  $N_i$  is the group of similar items to item  $j$ , and  $Sim(j,k)$  is the score of the similarity between the two items  $j$  and  $k$ .

The calculation of similarity among users/items constitutes a critical stage in neighborhood-based CF techniques, as it may severely decrease their accuracy and performance [19]. Several similarity metrics have been presented in the literature [20], among which cosine measure (COS) [21], Pearson correlation coefficient (PCC) [22] and Jaccard coefficient [23] are ones of the popular standard criteria typically adopted for finding most similar users or most similar items. PCC computes the similarity based on the linear correlation between two rating vectors of users/items. COS calculates the similarity by using the angle's cosine value between rating vectors. Jaccard similarity takes into account the number of common ratings between users/items and ignores the rating values. The choice of the similarity measure should be properly made on the basis of the target dataset [24]. To calculate the similarity measure between two users  $u$  and  $v$  respectively, these metrics are based on the following expressions:

$$sim^{PCC}(u,v) = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u) \cdot (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}} \tag{3}$$

$$sim^{COS}(u,v) = \frac{\sum_{i \in I_{u,v}} (r_{u,i}) \cdot (r_{v,i})}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i})^2} \cdot \sqrt{\sum_{i \in I_{u,v}} (r_{v,i})^2}} \tag{4}$$

$$sim^{Jaccard}(u,v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|}. \tag{5}$$

In these Equations,  $I_{u,v}$  denotes the items' set rated by users  $u$  and  $v$ ;  $\bar{r}_u$  represents the ratings' mean value of the user  $u$ , and  $r_{u,i}$  represents the  $u$ 's rating for the item  $i$ .  $I_u$  and  $I_v$  represent two items sets rated by users  $u$  and  $v$  respectively. On the other hand, the similarity among two items  $i$  and  $j$  is computed by involving users' ratings which have evaluated these two items:

$$sim^{PCC}(i,j) = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i) \cdot (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i)^2} \cdot \sqrt{\sum_{u \in U_{i,j}} (r_{u,j} - \bar{r}_j)^2}} \tag{6}$$

$$sim^{COS}(i,j) = \frac{\sum_{u \in U_{i,j}} (r_{u,i}) \cdot (r_{u,j})}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i})^2} \cdot \sqrt{\sum_{u \in U_{i,j}} (r_{u,j})^2}} \tag{7}$$

$$sim^{Jaccard}(i,j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|}, \tag{8}$$

where  $U_{i,j}$  accounts for the group of users who evaluated items  $i$  and  $j$ , and  $\bar{r}_i$  reflects the average value of ratings received by the item  $i$ .  $U_i$  and  $U_j$  refer to Users sets who rated items  $i$  and  $j$  respectively.

Nevertheless, the major shortcoming of memory-based CF is that these approaches may incur prohibitive computational costs (computation time of similarities among users or items), which augment with the growth in the number of users/items in the system [25]. However, they have become popular because of their uncomplicated implementing process, providing an understandability for the calculated predictions [17,26].

### 2.1.2. Model-Based CF

Despite the neighborhood-based CF techniques are simple to be implemented and effective in inferring unknown ratings of users, model-based CF approaches generally generate more precise predictions [18]. The basic idea of these types of techniques is the utilization of data mining and machine learning approaches for developing prediction models offline. Based on these models, RS predicts missing ratings in the user-item matrix [27]. During recent years, various model-based CF techniques have been developed, namely, Bayesian networks [28], neural networks [29], support vector machines [30], and very recently, fuzzy-based systems [31] and deep learning techniques [32]. Nevertheless, the Matrix factorization (MF) models [33] are regarded to be the state-of-the-art in RS due to their strengths in terms of accuracy and scalability [18]. MF algorithms use the high-level correlation among rows and columns (users and items) of a target user-item rating matrix for learning the users' and items' latent representations (also called latent factors) [34]. More precisely, each item  $i$  and each user  $u$  are respectively represented by  $k$ -dimensional latent factors, namely  $q_i \in R_k$  that represents  $k$ -characteristics of the item and  $p_u \in R_k$  that refers to the preference of the user for these characteristics. Formally, the rating score of a user  $u$  on item  $i$  is computed as follows [4,6]:

$$\hat{r}_{u,i} = p_u q_i^T. \quad (9)$$

To optimize latent factors which better predict  $\hat{r}_{u,i}$ , the following loss function must be minimized in such a way:

$$\min_{p^*, q^*} \sum_{(u,i) \in T} (r_{u,i} - p_u q_i^T)^2 + \beta (||p_u||^2 + ||q_i||^2), \quad (10)$$

where  $T$  represents the user-item  $(u, i)$  pairs for which real ratings  $r_{u,i}$  are observed in training set. And  $\beta$  is a defined regularization parameter which is used to limit the overfitting of the model. In general, the minimization of the loss function (Equation (6)) can be achieved with different techniques such as the Gradient-based or alternating least-squares [16].

Compared to neighbor-based CF techniques, the model-based CF return more accurate prediction results. Furthermore, storage requirements for these approaches are frequently less than those demanded by neighbor-based techniques [9]. This is because, in neighbor-based CF, all ratings are required to be loaded in memory to provide recommendations, while model-based CF involved the learned model, which is generally smaller than the original rating matrix [9]. Nevertheless, the model-building can requires more time and training data [35]. Besides, if novel users and/or products (items) are registered in the system, the new model should be trained multiple times to update it and maintain its accuracy [35].

## 2.2. Evaluation Metrics of CF

Evaluation represents an integral part of any system building process for proven its efficiency for the interest tasks [36]. To evaluate the performance of CF-based RS many evaluation measures have been used by research communities in RS [37]. These can be widely categorized into two main approaches—online and offline [37]. The first approach implies providing recommendations to the users and then querying them regarding how they assess the recommended items. Offline approach

does not involve real users’ interactions, rather part of the users’ historical data is exploited for training the system, whereas another part is utilized for testing the computed predictions. Online approach is considered the best evaluation method, due to its capacity of providing precise feedback of how pertinent the system is through real users [38]. Nevertheless, interactions with real users are mainly time-consuming, thus, many works have adopted an offline evaluation approach [20]. Table 2 presents some of the common evaluation metrics used in CF-based RS, their definitions, as well as their formulas.

**Table 2.** Evaluation metrics used in CF.

| Metrics                 | Definition  | Formula  | References    |
|-------------------------|---|--|---------------|
| Mean Absolute Error     | It measures the average of the absolute difference among the predicted ratings and true values.     | $MAE = \frac{1}{ T } \sum_{(u,i) \in T}  \hat{r}_{u,i} - r_{u,i} ,$ <p>where <math>r_{u,i}</math> refers to the real rating for user <math>u</math> over item <math>i</math> and <math>\hat{r}_{u,i}</math> is the predicted rating by a CF system, <math>T = \{(u,i)\}</math> denotes the set of user-item pairs for which the real ratings <math>r_{u,i}</math> are known.</p> | [22]          |
| Root Mean Squared Error | It emphasizes the contributions of the absolute errors between the predictions and the real values. | $RMSE = \sqrt{\frac{1}{ T } \sum_{(u,i) \in T} (\hat{r}_{u,i} - r_{u,i})^2}.$  | [18]          |
| Precision               | It computes the rate of the provided recommendations that are pertinent.                            | $Precision = \frac{ U_u \cap L_{rec} }{L_{rec}},$ <p>where <math>U_u</math> represents the number of all items used by the user <math>u</math> and <math>L_{rec}</math> is the list of recommended items.</p>  | [18]          |
| Recall                  | It computes the rate of recommendations that are provided.  | $Recall = \frac{ U_u \cap L_{rec} }{U_u}.$   | [18]          |
| ROC curve               | It amplifies the proportion of recommendations that are not preferred by the user.                  | Plots the true positive rate against the false positive rate.  | [18]          |
| Ranking Score           | It measures the quality of recommendations based on their rank position.                            | $rank(L_{rec}) = \sum_{j=1}^{ L_{rec} } \frac{\max(r_{(j)} - md, 0)}{2^{\frac{j-1}{\alpha}}},$ <p>where <math>r_{(j)}</math> is the item <math>i</math>'s rating in the rank <math>j</math>, <math>md</math> refers to the median rating and <math>\alpha</math> is the value of half-life decay.</p>  | [20]          |
| Click Trough Rate       | It computes the proportion of recommendations ultimately clicked                                    | $CTR = \frac{ L_{cons} }{ L_{rec} },$ <p>where <math>L_{cons}</math> is the list of consumed items</p>   | [39,40]       |
| Novelty                 | It computes the novelty of the provided recommendations   | $nov(L_{rec}) = \sum_{i \in L_{rec}} \min_{j \in L_{his}} dis(class(i), class(j)),$ <p>where <math>L_{his}</math> is the history's list of the user. <math>dis</math> is a distance measure, <math>class(i)</math> and <math>class(j)</math> represent the classes of items <math>i</math> and <math>j</math>, respectively.</p>   | [18,20]       |
| others                  | –   | –  | [18,20,22,36] |

### 2.3. Main Issues and Challenges on Standard CF Techniques

This subsection investigates the most common issues and challenges encountered in deploying CF-based RS and are considered important in the CF-based RS research.

### 2.3.1. Data Sparsity

Typically, there are a large number of missing ratings in the user-item interaction data, and the sparsity is frequently superior to 99% [41]. This is due to the difficulty that users encounter when they want to express their interests as numerical ratings on products [42], or because of the poor recommendation space's coverage [10]. This problem has a major negative influence on the effectiveness of CF approaches [43]. Due to the sparsity issue, it is likely that the similarities among users cannot be calculated, decreasing the effectiveness of CF. Alike when the similarities are calculable, they may be unreliable, since the information obtained is insufficient [43]. The review-based recommendation techniques discussed in Section 4 mitigate this problem in different manners.

### 2.3.2. Cold-Start

This issue takes place when novel users/items are added to the rating matrix. In such cases, CF methods are not able to provide these users with recommendations nor to recommend these items, since the system has not yet collected enough ratings about them [44]. To mitigate this problem, the content of user reviews can be combined with scalar ratings (Section 4).

### 2.3.3. Scalability

In a CF algorithm, it is expensive to calculate the users' similarity as the algorithm must search the entire database to determine the target user's potential neighbors [45]. Therefore, with a larger data set, algorithms require more resources like memory or computation power, which limit the algorithms to scale [46]. The practical solution to this issue may consist of using clustering CF approaches which search users in small size clusters rather than the complete database [47], or reducing dimensionality per singular value decomposition (SVD) [48], or combining content-analysis and clustering with CF techniques [49]. Another interesting solution for overcoming the scalability relies on the use of distributed computing mechanisms [50]. Different studies have incorporated the standard CF algorithms into a distributed computing engine to improve their computational performance on recommendation applications [50–52] through the use of Apache Hadoop or Spark, that are fast and practical frameworks for parallel large-scale data processing [53].

### 2.3.4. Limitations of Numerical Explicit Ratings

Typical CF methods suffer from a principal problem because of their dependency on users' numeric ratings as their unique source of user preference information [12]. However, the scalar rating information frequently lacking good enough semantic explanation to reflect the actual preferences of the user, thus greatly reducing the recommendation accuracy [9]. To address this problem, various recommendation approaches combine ratings and user reviews (see Section 4).

## 3. User Review Texts

The growth of electronic commerce has promoted users to write and share reviews expressing their opinion regarding items. Typically, these users' reviews are in free text form which expresses various dimensions or viewpoints of the experience that a user had for a given item [3]. They thus constitute a very valuable information source on preferences of users and may be used to learn fine-grained profiles of users and improve personalized suggestions. Chen et al. [6] identified different information elements that can be obtained from review texts and can be exploited by RS. Among these review elements, terms (words), aspects and opinions (sentiments) have been proved to be efficient for user modeling. In the following, we present these elements and briefly discuss the possibility of their usage in CF-based RS.

**Review Words:** The user review is in an unstructured textual form. The easiest way of mining it is to capture the most representative words. For instance, the TF-IDF weight measure [54] can be

utilized to indicate the relevance of each word in the review. The extracted review words may be used to compute the similarity among users, rather than utilizing numerical ratings in CF [55].

**Review Topics:** The topics refer to an item's aspects which a writer reviews in its review. For instance, in the review phrase: "The camera's battery life is superb" the mentioned topics include the camera and its battery life. There are various methods for topics detection in reviews, namely, frequency-based, syntax-based, Conditional Random Fields [56], and topic modeling approaches like Latent Dirichlet Allocation (LDA) [57], Latent Semantic Analysis (LSA) [58], or Probabilistic Latent Semantic Analysis (PLSA) [59]. Review topics can then be used to improve real ratings in standard CF [34]. They can also be combined with latent factors in model-based CF [60] and with the similarity measure in neighbor-based CF [61].

**Overall Opinions:** They represent the sentiment orientation (i.e., positive or negative) of the user towards reviewed items. Generally, the overall opinion may be deducted by regrouping all opinion words' sentiments in the reviews or by applying a coarse-grained sentiment analysis method based on supervised [62,63], semi-supervised [64] or unsupervised machine learning techniques [65]. The extracted overall opinions can be transformed into scalar ratings, that can be useful to augment CF techniques performance [66–68].

**Aspect Opinions:** They represent the detailed opinions about an item's particular characteristics. For example, the review phrase "The waiters' attitude is great". discloses a positive opinion on the service aspect. In general, review aspects may reference to a distinct thing like the product itself or one of its attributes ("attitude of waiters" rather than "service"). The typical techniques to feature extraction include linguistics-based methods and statistical methods [69–72], or structured models, like Conditional Random Fields (CRF) [73], Hidden Markov Models (HMM), and their variations [73,74]. The identification of opinions associated with aspects (features) is then made through word distance or pattern mining [69,75]. Alternatively, an SVM or LDA classifiers can be utilized for identifying the aspect opinions (aspect, sentiment pairs) [6]. In Reference [76], the aspect sentiments were utilized for calculating user similarities in order to cluster users in CF. In Reference [7] they were used to identify user similarities, and then incorporated into standard user-based CF.

#### 4. CF Techniques Based on User Review Texts

Recently, many attempts have been made for integrating the precious information incorporated in user reviews into the recommendation task [6]. This section summarizes a list of recent works on review-based CF recommender systems (Tables 3–6). Particularly, these works can be classified into three principal techniques, namely, techniques based on words, on topics and finally on opinions.

##### 4.1. Techniques Based on Review Words

These techniques use the review words by factorizing them into CF. For instance, Terzi et al. [55] proposed a modification of the user-based technique which computes the similarities among users based on text reviews' similarities, rather than ratings. More precisely, the similarity between two users is calculated by measuring the similarity among reviews' words of these two users for every co-reviewed item. The computed similarities scores are then utilized as a weight in the rating prediction phase.

Kim et al. [77] proposed a Convolutional Matrix Factorization (ConvMF) model, which utilizes reviews text as complementary information. Firstly, this model utilizes convolutional operations and word embedding for capturing the items' latent characteristics from their review texts. After that, the inferred latent features are integrated into a matrix factorization model to compute the users' ratings on target items.

Zheng et al. [78] proposed a Deep Cooperative Neural Networks (DeepCoNN) model which uses two parallel convolutional neural networks (CNNs) and a word embedding method for capturing latent representations for the all reviews' words associated to a target user and item. To perform the

prediction task, the model concatenates the user and item representations and then transmits it to a regression layer involving a Factorization Machine (FM) technique.

Similar to DeepConn [78], the model developed by Chen et al. [79] (called NARRE) uses CNNs to derive latent embeddings of users and items from review texts. Different from DeepConn, it scores reviews through an attention network to distinguish their contribution when learning the latent embeddings. To predict missing ratings, NARRE uses attention scores with user latent rating factors and then incorporates them into an extended MF.

The work in Reference [80] fused the ratings and review information in a unified model. The model exploits CNNs and an attention mechanism to learn the relevant latent features by considering their related reviews. Through a rating-based component, the model constructs latent rating embeddings for users and items from the interaction matrix. To derive the final rating score, the learned content features and latent rating embeddings are integrated into a Factorization Machine (FM).

Very recently, Liu et al. [81] presented a Hybrid neural recommendation model (called HRDR) to capture user and item embeddings from reviews and ratings. Firstly, the rating representations are obtained from rating data by using a Multilayer Perceptron (MLP) network. Then, CNNs with an attention mechanism are used to derive review-based representations where each review is associated with an informativeness score. Finally, a MF is used to compute users' ratings on items based on their latent ratings, review features and ID-embeddings.



Table 3. Related works on techniques based on review words.

| Citation                                   | User/Item Profile   | Recommending Method                               | Tested Datasets   | Main Contribution  | Accuracy Performance                       |                   |   |
|--|---|---|---|--|--|-------------------|---|
|  |   |   |   |  | Product Reviews                            | Achieved Accuracy | Accuracy of CF Baselines  |
| Terzi et al. [55]<br>(Text-based user-kNN) | Review Words  | User-based CF                                     | Rottentomatoes (movies), Amazon (Audio CDs)   | Improve accuracy (RMSE)  | Audio CDs                                  | 1.1092            | User-knn: 1.1190<br>Item-knn: 1.1130<br>SVD++ [82]: 1.1099<br>BMF [33]: 1.1105  |
| Kim et al. [77]<br>(ConvMF)                | Latent ratings and item review words  | CNN with Probabilistic Matrix Factorization (PMF) | Amazon (Instant Video), MovieLens (movies)  | Enhance the rating prediction accuracy (RMSE)                            | Instant Video                              | 1.1337            | PMF [83]: 1.4118<br>CTR [84]: 1.5496  |
| Zheng et al. [78]<br>(DeepCoNN)            | Latent factors from review words  | CNN with Factorization Machine                    | Yelp (restaurants), Amazon (Musical instruments), Beer (beers)  | Improve prediction accuracy (MSE), Alleviate the sparsity problem        | Musical Instruments, restaurants and beers | 0.994             | MF [33]: 1.292<br>PMF [83]: 1.256<br>CTR [84]: 1.112  |
| Chen et al. [79]<br>(NARRE)                | Latent factors from ratings, and latent factors based on reviews              | CNN with MF                                       | Amazon (Toys_and_Games, Kindle_Store, and Movies_and_TV), Yelp (businesses)   | Increase prediction accuracy (RMSE), Interpretability in recommendations | Kindle Store                               | 0.7783            | PMF [83]: 0.9914<br>NMF [85]: 0.9023<br>SVD++ [82]: 0.7928<br>HFT [60]: 0.7917<br>DeepCoNN [78]: 0.7875                                       |
| Wu et al. [80]<br>(CARL)                   | Latent feature ratings, latent factors from review words                      | CNN and Factorization machine                     | Amazon (Musical Instruments, Office Products, Digital Music, Video Games, and Tools Improvement), RateBeer (Beer), Yelp (Restaurants) | Augment rating prediction performance (MSE)                              | Musical Instruments                        | 0.776             | PMF [83]: 1.401<br>ConvMF [77]: 0.991<br>DeepCoNN [78]: 0.814<br>RBLT [86]: 0.815   |
| Liu et al. [81]<br>(HRDR)                  | Explicit features from ratings, semantic features from reviews, ID embeddings | CNN with MF                                       | Yelp 2013 and Yelp 2014 (yelp.com), Amazon (Video games and Gourmet food)   | Augment recommendation accuracy (RMSE)                                   | Video games                                | 1.011             | PMF [83]: 1.139<br>HFT [60]: 1.073<br>CTR [84]: 1.071<br>JMARS [87]: 1.064<br>ConvMF+[77]: 1.073<br>DeepCoNN [78]: 1.063<br>NARRE [79]: 1.055 |

Table 4. Related works on techniques based on review topics.

| Citation                        | User/Item Profile                            | Recommending Method                  | Tested Datasets  | Main Contribution  | Accuracy Performance         |                   |  |
|---------------------------------|--|--------------------------------------|--|--|------------------------------|-------------------|--|
|                                 |  |                                      |  |  | Product Reviews              | Achieved Accuracy | Accuracy of CF Baselines                 |
| McAuley and Leskovec [60] (HFT) | Latent ratings merged with topic factors     | Hidden Factors as Topics (HFT)       | Amazon (movies, books, etc.), Beeradvocate and Ratebeer (wines, beers), Yelp (restaurants), etc. | Improves rating prediction accuracy (MAE), Tackle rating sparsity issue                                | 26 Amazon product categories | 1.329             | LFM [33]: 1.423                          |
| Tan et al. [86] (RBLT)          | Latent topic opinions, latent rating factors | Matrix Factorization                 | Amazon (26 datasets [60])  | Prediction accuracy improvement (MSE), Alleviate data sparsity problem                                 | Video Games                  | 1.462             | LFM [33]: 1.487                          |
| Bao et al. [88] (TopicMF)       | Latent factors associated with topic factors | Topic Matrix factorization (TopicMF) | Amazon (arts, automotive, baby, beauty, etc.) [60]   | Enhance prediction accuracy (MSE)  | 22 Amazon product categories | 1.3468            | PMF [83]: 1.5585<br>SVD++ [82]: 1.4393   |
| Cheng et al. [89] (ALFM)        | Latent topics, latent rating factors         | Matrix Factorization                 | Amazon (26 datasets [60]), Yelp (businesses)   | Improve prediction accuracy RMSE, Alleviate data sparsity problem, Interpretability in recommendations | Musical Instruments          | 0.893             | BMF [33]: 1.004                          |
| Chin et al. [90] (ANR)          | Latent aspect ratings and aspect importance  | Aspect-based Neural Recommender      | Amazon (24 datasets [60]), Yelp (businesses)   | Prediction accuracy improvement (MSE)  | Instant Video                | 1.009             | DeepCoNN [78]: 1.178<br>ALFM [89]: 1.075 |

**Table 5.** Related works on techniques based on review sentiments.

| Citation                      | User/Item Profile                            | Recommending Method                     | Tested Datasets   | Main Contribution   | Accuracy Performance |                   |  |
|-------------------------------|--|---|---|---|----------------------|-------------------|--|
|                               |  |   |   |   | Product Reviews      | Achieved Accuracy | Accuracy of CF Baselines                                 |
| Poirier et al. [66]           | Ratings from opinion classification          | Item-based CF                           | Flixster (movies)   | Overcoming the cold-start issue (RMSE)                                  | Movies               | 0.898             | user-based CF: 0.897                                     |
| Zhang et al. [91] (EFM)       | Ratings and aspect sentiment scores          | Factorization model                     | Yelp (businesses), Dianping (restaurants)   | Improve prediction accuracy (RMSE)                                      | Businesses           | 1.212             | PMF [83]: 1.253<br>NMF [85]: 1.248                       |
| Diao et al. [87] (JMARS)      | Latent ratings and aspects' sentiment scores | Probabilistic matrix factorization      | IMDB (movies)   | Prediction accuracy increasing and address the cold start problem (MSE) | Movies               | 4.97              | PMF [83]: 5.99   |
| Ma et al. [7] (UPCF)          | Ratings and aspects' opinion ratings         | User-based CF                           | Dianping (restaurants)  | Accuracy increasing (RMSE), Deal with sparsity problem                  | Restaurants          | 0.7707            | User-based CF: 0.7902<br>item-based CF: 0.8199           |
| Musto et al. [92] (Multi-U2U) | Aspects' opinion scores                      | Multi-criteria based user/item-based CF | Yelp (restaurants), TripAdvisor (hotels), Amazon (Video Games)  | Increase prediction accuracy (MAE)                                      | Video Games          | 0.6276            | User-based CF: 0.9789<br>Item-based CF: 0.9679           |
| Shen et al. [68] (SBFM)       | Ratings with Reviews' sentiment scores       | Probabilistic matrix factorization      | Amazon (Patio_lawn_and_garden, Office products, Amazon instant video, Baby, Tools and home improvement, Beauty, Cellphones and accessories, Clothing and accessories) | Prediction accuracy improvement (Normalized RMSE)                       | Beauty               | 0.2898            | MF [33]: 0.3411<br>PMF [83]: 0.3338<br>HFT [60]: 0.3085  |
| Da'u et al. [93] (AODR)       | Ratings and aspect-sentiment scores          | Tensor Factorization                    | Amazon (Musical Instruments, Automotive, Instant Video), Yelp (businesses)  | Augment Rating prediction and address data sparseness (RMSE, MAE)       | Instant Video        | 0.7990            | MF [33]: 0.9583<br>HFT [60]: 0.8172<br>RBLT [86]: 0.8061 |

Table 6. Characteristics of review-based approaches.

| Category        | Approach                 | Characteristics               |   |  |  |   |   |   |  |                                  |  |                                |
|-----------------|--------------------------|-------------------------------|---|--|--|---|---|---|--|----------------------------------|--|--------------------------------|
|                 |                          | External NLP Tool Is Required | Consider Texts of Review as Simple Bag of Words | Static and Independent Vectors of Users and/or Items | Integrate Ratings in the Modeling Process of Reviews | Correlation between the Review’s Features | Emphasize the Pertinent Reviews or Parts of the Reviews | One-to-One Mapping (Latent Ratings and Latent Features) | Uses User-Specific Opinions on Item’s Features | Less Explainable and Informative | Powerful Representation Learning Abilities | Complex Implementation Process |
| Word-based      | Text-based user-kNN [55] | •                             | •   | •  |  |   |   |   |  |                                  |  |                                |
|                 | ConvMF [77]              |                               |   | •  |  | •   |   |   |  | •                                | •  |                                |
|                 | DeepCoNN [78]            |                               |   | •  |  | •   |   |   |  | •                                | •  |                                |
|                 | NARRE [79]               |                               |   | •  |  | •   | •   |   |  |                                  | •  | •                              |
|                 | CARL [80]                |                               |   |  |  | •   | •   |   |  |                                  | •  | •                              |
|                 | HRDR [81]                |                               |   | •  | •  | •   | •   |   |  | •                                | •  | •                              |
| Topic-based     | HFT [60]                 |                               | •   | •  |  |   |   |   |  |                                  |  |                                |
|                 | RBLT [86]                |                               | •   | •  | •  |   |   |   |  |                                  |  |                                |
|                 | TopicMF [88]             |                               | •   | •  |  |   |   |   |  |                                  |  |                                |
|                 | ALFM [89]                |                               | •   |  |  |   |   |   |  |                                  |  |                                |
|                 | ANR [90]                 |                               |   |  |  | •   | •   |   |  |                                  | •  | •                              |
| Sentiment-based | Poirier et al. [66]      | •                             | •   | •  |  |   |   |   |  |                                  | •  |                                |
|                 | EFM [91]                 | •                             | •   | •  |  |   |   |   |  |                                  |  |                                |
|                 | JMARS [87]               | •                             | •   |  |  |   |   |   |  |                                  |  |                                |
|                 | UPCF [7]                 | •                             |   | •  |  | •   |   |   |  |                                  |  |                                |
|                 | Multi-U2U [92]           | •                             | •   | •  |  |   |   |   |  |                                  |  |                                |
|                 | SBFM [68]                |                               |   | •  | •  |   |   |   |  |                                  | •  |                                |
|                 | AODR [93]                | •                             |   |  | •  | •   | •   |   |  | •                                | •  | •                              |

#### 4.2. Techniques Based on Review Topics

This type of technique extracts aspects from reviews and combines them with ratings for generating recommendations. For example, McAuley and Leskovec [60] proposed a Hidden Factor and Topic (HFT) framework that fuses ratings with review topics. Firstly, it models reviews with the LDA-based topic model and ratings with standard MF. Then, a Softmax transformation function is used for incorporating the latent topics into the learning phase of the latent features model. Based on the trained model, the final rating scores are computed.

In the same way as McAuley and Leskovec [60], the model proposed by Tan et al. [86] (called RBLT) utilizes MF for modeling rating scores and LDA for representing the text of reviews. In their model, items are represented as topical distribution, and the topics in elevated rating reviews are repeated for augmenting their importance. Alike, users are represented in a similar topical space by their numerical ratings. To perform the rating prediction task, the item and user representations are fused into a latent factorization model.

Based on the fact that the LDA technique cannot model the compound topics' distribution, authors of [88] extended HFT [60] by proposing the TopicMF framework. TopicMF captures topics from user review text based on non-negative MF, and utilizes a MF technique for factorizing rating matrix into latent user/item features. For rating prediction, a transform action function is used to join the topic features with the matching latent user/item features.

More recently, Cheng et al. [89] proposed an Aspect-Aware Latent Factor Model (ALFM) that leverages an Aspect-aware Topic Model (ATM) for modeling aspect-level user/item representations as distributions of composite topics, each of which is represented by a set of words. In ALFM, the resulted representations from ATM are fused with latent rating factors to estimate the missing ratings based on the MF model.

Chin et al. [90] proposed an Aspect-based Neural Recommender (ANR) that uses a neural network for estimating the latent aspects ratings and latent aspects importance. The latent aspects ratings are derived through a weighted sum of all the words' embedding in the reviews. The latent aspects importance is inferred by using a shared similarity among each pair of the user's item's latent aspects ratings. Finally, the overall rating for any user-item pair is inferred by combining their associated aspects ratings with aspects importance into a modified Latent Factor Model (LFM).

#### 4.3. Techniques Based on Review Sentiments

Research works in this area use the user's expressed sentiment on the item itself or on its different aspects in reviews, to boost the rating prediction task. For instance, Poirier et al. [66] transform reviews into overall sentiment scores based on a machine learning method. To do that, reviews vectors fused with users' real ratings are exploited for training a Naive Bayes model on negative and positive classes. This learned model is then utilized for deducting ratings from novel reviews. To predict ratings, the review-based ratings are used for constructing a rating matrix that is integrated into the traditional neighbor-based CF techniques.

Differently, in Reference [91] an Explicit Factor Model was developed to transform user reviews into aspect-sentiment pairs. Based on phrase-level sentiment analysis, it constructs two matrices, namely, user-aspect attention and item-aspect quality, which are simultaneously decomposed with the rating matrix for performing rating prediction in a MF-based model.

The model proposed by Diao et al. [87] (called JMARS), utilizes the relationship between review aspects, opinions and ratings to conduct CF. It exploits the Dirichlet-Multinomial technique for capturing the reviews' word distribution and a MF for generating the aspects ratings which are fused with latent factors to compute the final rating scores.

On the other hand, Ma et al. [7] have presented a user-preference-based CF that integrates aspect-level information to reflect user interests from reviews. Specifically, two metrics for aspect interests have been proposed, namely aspect need and aspect importance for reflecting the differences of opinions to aspects and the aspect relationship to explicit rating, respectively. Based on these

measures, the authors compute the similarity between users, which is then incorporated into memory-based CF to further recommendations.

Musto et al. [92] developed multi-criteria user- and item-based CF techniques that integrate opinion information of reviews' aspects. For user/item-based cases, the authors present aspect-based item/user distances, which utilize the sentiment ratings deduced from reviews' aspects. The similarity between users or items is then computed as the inverse of the proposed distances, and ratings are calculated using the standard CF model. In the paper, the authors use the SABRE engine [94] for performing the aspect extraction task.

Shen et al. [68] developed a sentiment-based MF model that incorporates reviews' sentiments. To infer the review's overall sentiments scores, this model sums the sentiment score of each keyword in the target review based on the score obtained from a constructed sentiment dictionary. To perform rating prediction, these sentiment scores are converted into real values and then fused with the users' explicit ratings into an extended probabilistic MF.

In a recent work [93], the authors proposed a unified model to integrate aspects opinion information into CF. The model uses a multichannel CNN that involves word embedding and POS tag embedding layers for extracting review aspects. It regroups aspects by using an LDA technique and then exploits a lexicon approach for building the aspects rating matrices. The aspects ratings are then weighted based on a tensor factorization method and integrated with a rating matrix into an LFM for predicting final ratings.

## 5. Practical Benefits of Review Incorporation

From Tables 3–5 and 7, we can see that all works on review-based CF algorithms have proven their advantages compared to the traditional CF recommending approaches. This section discusses the practical benefits of these review-incorporated techniques on two main issues, namely, rating sparsity and rating prediction improvement.

### 5.1. Rating Sparsity

As indicated in Section 2, the lack of pertinent data like sparsity considerably reduces the efficiency of the CF techniques [7]. To tackle this problem, researchers have explored user reviews in different ways (see Tables 3–5 and 7):

The works proposed in References [60,86–89] have demonstrated the capacity of their approaches to mitigating the rating sparsity issue. These works exploit review topics (aspects) for enriching the latent factor model. They extract aspects from review texts using topic models and learn latent features from ratings using MF methods. Then, the latent topics and latent factors are combined in a way for boosting prediction performance. For example, HFT [60] uses a defined transform function to learn the latent factors and latent topics together. JMARS model [87] leverages a one-to-one matching among the latent factors and the learned latent aspects for determining the final ratings. Bao et al. [88] fuse aspects in reviews with latent factors in a user-item rating matrix by exploiting a transform function. Tan et al. [86] use a linear combination between them for building the final users/items representations which are then used in the rating prediction task. Cheng et al. [89] extract reviews' topics and associate them with aspects, and then use an extended latent factor model to enrich latent ratings with aspects.

Poirier et al. [66] show that user reviews can be converted into text-based ratings and then used to replace the user explicit ratings in the CF process. This approach first infers opinion ratings from reviews based on a machine learning model and then executes a neighbor-based CF method. Therefore, this work has proved its ability to mitigate the rating sparsity issue by inferring ratings from review texts.

On the other hand, Ma et al. [7] leverage review text for capturing the weights preference which the target user assigns to different aspects. To derive the aspect preferences, all the user's reviews are used, making easy the similarities' computation among each users' pairs, no importance how a number of items they frequently rate, that can mitigate the data sparseness issue.

**Table 7.** Comparisons based on Sparsity situations.

| <b>Approach</b> | <b>Condition on Datasets (Level of Sparsity)</b>   | <b>Improvement in Accuracy Compared to Baselines</b>   |
|-----------------|--|--|
| UPCF [7]        | Dianping (Data-5) with #Reviews of each user: 5–9  | MSE of HFT [60] < MSE of User-based CF   |
| HFT [60]        | Amazon (movies) with #Reviews for each user/product: 1–10  | MSE of LFM [33] – MSE of HFT > 0   |
| RBLT [86]       | Amazon (26 datasets) with #Reviews for each user/item: 1–10  | MSE of (LFM [33]/HFT [60]) – MSE of RBLT > 0   |
| JMARS [87]      | IMDB #training reviews for each user/movie: 1–100  | MSE of HFT [60] – MSE of JMARS > 0   |
| ALFM [89]       | Amazon (24 item categories) #reviews for each user/item: 1–10  | RMSE of (BMF [33]/HFT [60]/RBLT [86]) – RMSE of ALFM > 0   |
| ConvMF [77]     | MovieLens-1m: 7 sub-datasets of different densities (0.93%; 1.39%; 1.86%; 2.32%; 2.78%; 3.25%; 3.71%)            | RMSE of ConvMF < RMSE of (PMF [83]/CTR [84])   |
| DeepCoNN [78]   | Three datasets: Yelp, Beer, and Amazon (Music Instruments) with #training reviews for each user/item: 1–5        | MSE of MF [33] – MSE of DeepConn > 0   |
| AODR [93]       | Amazon (Musical Instruments, Automotive, Instant Video) and Yelp datasets with #reviews for each user/item: 1–10 | RMSE of (BMF [33]/HFT [60]/RBLT [86]) – RMSE of AODR > 0<br>MAE of (BMF [33]/HFT [60]/RBLT [86]) – MAE of AODR > 0 |

Differently, Kim et al. [77], Zheng et al. [78] and Da' u et al. [93] have demonstrated the capacity of their approaches to alleviating the sparsity problem by using rich semantic features extracted from review words through CNNs. Specifically, these studies confirmed that the use of CNN helps adjust latent ratings by efficaciously representing contextual features of user/item review texts when the rating data is sparse.

## 5.2. Rating Prediction Improvement

A lot of works (Section 4) propose to incorporate user review texts for improving the traditional CF techniques (Section 2). These works can be classified into two main categories. The first one focuses on modifying the standard CF techniques to integrate implicit scores inferred from review texts to adjust explicit ratings and get more reliable and fine-grained ratings. For instance, the authors of References [60,86–89,91] have presented different modified version of the standard latent factor model, namely, HFT, RBLT, JMARS, TopicMF, ALFM and EFM models for improving numerical ratings by aligning them with latent topics in reviews. The works in References [77,79–81,93] have improved the real ratings in traditional latent factor by fusing them with latent feature vectors inferred from review words through an integrated CNN architecture. On the other hand, in Reference [7] the traditional user similarity in neighborhood-based CF recommenders has been improved by considering the users' aspect preference vectors inferred from reviews. Moreover, in Reference [68], the standard probabilistic MF has been improved through an adjustment of its real ratings by the sentiment scores inferred from reviews.

The second category focuses on replacing the explicit user ratings in standard CF with implicit ones generated from review texts. For example, the text-based ratings inferred from reviews can replace explicit ratings in neighbor-based CF approaches [66,92]. The review words can be used to improve the traditional user-kNN similarity in memory-based CF techniques [55]. The users' and items' latent embeddings obtained by CNNs from reviews can be used as features in an LFM to conduct rating predictions [78].

These existing review-incorporated works have proven their efficiency in exploiting user review texts (see summaries in Tables 3–5). For instance, in Reference [55] the extended user-based CF approach exploiting text-based ratings has been proven to generate more accurate predictions than traditional ratings-based approaches. The item-based CF approach exploiting reviews' ratings [66] has shown a comparable precision accuracy to standard CF which is based on explicit ratings. In References [7,92], the neighborhood-based CF technique based on inferred sentiment scores has been shown to provide results superior to the traditional memory-based CF approaches. On the other hand, the modified latent factor models that fuse real ratings with review-based ratings have proven to be more precise than the traditional models which only leverage real ratings [60,68,77,79–81,86–89,91,93]. This is due to the rich information of user interests and item characteristics contained in reviews, that could be practical complementary to numerical ratings.

Furthermore, certain works have compared different review-based CF methods. The neural network techniques [68,77–81,90,93] usually outperform methods that rely on CF with topic modeling [60,86–88,91] because of the robust representation capacity of neural network architectures, that can capture rich semantic features from review texts for representing users and items. However, techniques relying on topic modeling lose the deep textual characteristics through this coarse-grained text mining method.

On the other hand, we realize that the techniques [79–81,90,93] leveraging attention network usually outperform techniques without attention [77,78]. This is due to the usage of the attention mechanism, that allows capturing the more significant features in reviews and consequently provide a way for deriving users' and items' representations more precisely.



## 6. Conclusions

Nowadays, due to the occurrence of modern text mining techniques, much effort has been devoted to incorporating review texts into the recommending task. Different types of review elements, like review words, review topics, and review opinions have been utilized for augmenting the classical rating-based CF models because they allow to represent more accurately items and user's interests. In this paper, we survey existing review-based CF recommender systems and categorized them into three main systems, namely, systems based on words, on topics and finally on sentiments. For each one, we discuss how user review texts have been exploited to enrich rating profiles, and derive feature preference. We also discuss the practical benefits of these review-based recommending systems in terms of alleviating the rating sparsity and augmenting the prediction accuracy. In spite of the remarkable progress in the review-based CF RS research area, we can notice through our survey of different review-based approaches, that further works are needed. For instance, fusing various review-based CF RS might be further efficient than using a single system to predict users' preferences; another area of future work may be relying on the usage of advanced text mining approaches for identifying more complex relatedness among reviews and ratings.

**Author Contributions:** Conceptualization, M.S., A.A.L., S.M.; writing—original draft preparation, M.S.; writing—review and editing, M.S., A.O., A.A.L.; supervision, A.A.L., S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## References

1. Abdullah, L.; Ramli, R.; Bakodah, H.O.; Othman, M. Developing a causal relationship among factors of e-commerce: A decision making approach. *J. King Saud Univ. Comput. Inf. Sci.* **2019**. [\[CrossRef\]](#)
2. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–132. [\[CrossRef\]](#)
3. Sundermann, C.V.; Domingues, M.A.; Sinoara, R.A.; Marcacini, R.M.; Rezende, S.O. Using Opinion Mining in Context-Aware Recommender Systems: A Systematic Review. *Information* **2019**, *10*, 42. [\[CrossRef\]](#)
4. Francesco, R.; Rokach, L.; Shapira, B. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 1–34.
5. Yang, Z.; Xu, L.; Cai, Z.; Xu, Z. Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. *Knowl. Based Syst.* **2016**, *77*, 74–88. [\[CrossRef\]](#)
6. Li, C.; Chen, G.; Wang, F. Recommender systems based on user reviews: The state of the art. *User Model. User Adapt. Interact.* **2015**, *25*, 99–154.
7. Yue, M.; Chen, G.; Wei, Q. Finding users preferences from large-scale online reviews for personalized recommendation. *Electron. Commer. Res.* **2017**, *17*, 3–29.
8. He, X.; Chen, T.; Kan, M. Y.; Chen, X. Trirank: Review-aware explainable recommendation by modeling aspects. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015.
9. Han, H.; Huang, M.; Zhang, Y.; Bhatti, U.A. An Extended-Tag-Induced Matrix Factorization Technique for Recommender Systems. *Information* **2018**, *9*, 143. [\[CrossRef\]](#)
10. Alshammari, G.; Jorro-Aragoneses, J.L.; Polatidis, N.; Kapetanakis, S.; Pimenidis, E.; Petridis, M. A switching multi-level method for the long tail recommendation problem. *J. Intell. Fuzzy Syst.* **2019**, *37*, 7189–7198. [\[CrossRef\]](#)
11. Su, J.-H.; Chang, W.; Tseng, V.S. Effective social content-based collaborative filtering for music recommendation. *Intell. Data Anal.* **2017**, *21*, S195–S216. [\[CrossRef\]](#)
12. Zhang, Z.; Zhang, D.; Lai, J. *urCF: User Review Enhanced Collaborative Filtering*; AMCIS: Bubendorf, Switzerland, 2014.
13. Nikolay, A.; Ghose, A.; Ipeirotis, P.G. Deriving the pricing power of product features by mining consumer reviews. *Manag. Sci.* **2011**, *57*, 1485–1509.

14. María, H.; Cantador, I.; Bellogín, A. A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Model. User Adapt. Interact.* **2019**, *29*, 381–441.
15. Gediminas, A.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749.
16. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 29–39. [[CrossRef](#)]
17. 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](#)]
18. Aggarwal, C.C. *Recommender Systems*; Springer International Publishing: Cham, Switzerland, 2016; Volume 1.
19. Christian, D.; Karypis, G. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 107–144.
20. Silveira, T.; Zhang, M.; Lin, X.; Liu, Y.; Ma, S. How good your recommender system is? A survey on evaluations in recommendation. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 813–831. [[CrossRef](#)]
21. Greg, L.; Smith, B.; York, J. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80.
22. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [[CrossRef](#)]
23. Georgia, K.; Bercovitz, B.; Garcia-Molina, H. FlexRecs: Expressing and combining flexible recommendations. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, Providence, RI, USA, 29 June–2 July 2009.
24. Masoumeh, R.; Sohrabi, M.K. Providing effective recommendations in discussion groups using a new hybrid recommender system based on implicit ratings and semantic similarity. *Electron. Commer. Res. Appl.* **2020**, *40*, 100938.
25. Jon, H.; Konstan, J.A.; Riedl, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.* **2002**, *5*, 287–310.
26. Kumar, R.S.; Pateriya, R.K. Accelerated singular value decomposition (asvd) using momentum based gradient descent optimization. *J. King Saud Univ. Comput. Inf. Sci.* **2018**. [[CrossRef](#)]
27. Yang, Z.; Wu, B.; Zheng, K.; Wang, X.; Lei, L. A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access* **2016**, *4*, 3273–3287. [[CrossRef](#)]
28. Su, X.; Taghi, M.K. Collaborative filtering for multi-class data using belief nets algorithms. In Proceedings of the 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), Arlington, VA, USA, 13–15 November 2006.
29. Ruslan, S.; Mnih, A.; Hinton, G. Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, ON, USA, 20–24 June 2007.
30. Xia, Z.; Dong, Y.; Xing, G. Support vector machines for collaborative filtering. In Proceedings of the 44th Annual Southeast Regional Conference, Melbourne, FL, USA, 10–12 March 2006.
31. Wang, S.-T.; Li, M.-H. Mobile Phone Recommender System Using Information Retrieval Technology by Integrating Fuzzy OWA and Gray Relational Analysis. *Information* **2018**, *9*, 326. [[CrossRef](#)]
32. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 1–38. [[CrossRef](#)]
33. Yehuda, K.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37.
34. Qiu, L.; Gao, S.; Cheng, W.; Guo, J. Aspect-based latent factor model by integrating ratings and reviews for recommender system. *Knowl. Based Syst.* **2016**, *110*, 233–243. [[CrossRef](#)]
35. Su, X.; Taghi, M.K. A survey of collaborative filtering techniques. In *Advances in Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2009.
36. Rabiou, I.; Salim, N.; Da'u, A.; Osman, A. Recommender System Based on Temporal Models: A Systematic Review. *Appl. Sci.* **2020**, *10*, 2204. [[CrossRef](#)]
37. Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P. *Recommender Systems Handbook*; Springer: Berlin, Germany, 2011; pp. 1–34.

38. Garg, D.; Gupta, P.; Malhotra, P.; Vig, L.; Shroff, G. Sequence and time aware neighborhood for session-based recommendations: Stan. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019.
39. Negar, H.; Mobasher, B.; Burke, R. Context adaptation in interactive recommender systems. In Proceedings of the 8th ACM Conference on Recommender Systems, Foster City, SV, USA, 6–10 October 2014.
40. Carlos, G.-U.A.; Hunt, N. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manag. Inf. Syst.* **2015**, *6*, 1–19.
41. Yue, S.; Larson, M.; Hanjalic, A. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.* **2014**, *47*, 1–45.
42. Leung Cane, W.K.; Chan, S.C.F.; Chung, F. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In Proceedings of the ECAI 2006 Workshop on Recommender Systems, Riva del Garda, Italy, 28–29 August 2006.
43. Manos, P.; Plexousakis, D.; Kutsuras, T. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *International Conference on Trust Management*; Springer: Berlin/Heidelberg, Germany, 2005.
44. Shah, K.; Ali, Z.; Ullah, I. Recommender systems: Issues, challenges, and research opportunities. In *Information Science and Applications (ICISA) 2016*; Springer: Singapore, 2016; pp. 1179–1189.
45. Jiang, J.; Lu, J.; Zhang, G.; Long, G. Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop. In Proceedings of the 2011 IEEE World Congress on Services, Washington, DC, USA, 4–9 July 2011.
46. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May, 2001.
47. Lee, O.J.; Hong, M.S.; Jung, J.J.; Shin, J.; Kim, P. Adaptive collaborative filtering based on scalable clustering for big recommender systems. *Acta Polytech. Hung.* **2016**, *13*, 179–194.
48. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. *Application of Dimensionality Reduction in Recommender System—A Case Study*; No. TR-00-043; Minnesota University, Department of Computer Science: Minneapolis, MN, USA, 2000.
49. Shahabi, C.; Banaei-Kashani, F.; Chen, Y.S.; McLeod, D. Yoda: An accurate and scalable web-based recommendation system. In Proceedings of the International Conference on Cooperative Information Systems, Trento, Italy, 5–7 September 2001.
50. Sun, J.; Wang, Z.; Luo, X.; Shi, P.; Wang, W.; Wang, L.; Wang, J.H.; Zhao, W. A parallel recommender system using a collaborative Filtering algorithm with correntropy for social networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 91–103. [[CrossRef](#)]
51. Christos, S.; Papadatos, G.B.; Varlamis, I. Optimizing parallel collaborative filtering approaches for improving recommendation systems performance. *Information* **2019**, *10*, 155.
52. Riyaz, P.A.; Varghese, S.M. A scalable product recommendations using collaborative filtering in hadoop for bigdata. *Procedia Technol.* **2016**, *24*, 1393–1399. [[CrossRef](#)]
53. Oussous, A.; Benjelloun, F.Z.; Lahcen, A.A.; Belfkih, S. Big Data technologies: A survey. *J. King Saud Univ. Comput. Inf. Sci.* **2018**, *30*, 431–448. [[CrossRef](#)]
54. Gerard, S.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523.
55. Terzi, M.; Rowe, M.; Ferrario, M.A.; Whittle, J. Text-based user-knn: Measuring user similarity based on text reviews. In Proceedings of the International Conference on User Modeling, Adaptation, and Personalization, Aalborg, Denmark, 7–11 July 2014.
56. Niklas, J.; Gurevych, I. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; pp. 1035–1045.
57. Susana, Z.; Vulić, I.; Moens, M. Latent Dirichlet allocation for linking user-generated content and e-commerce data. *Inf. Sci.* **2016**, *367*, 573–599.
58. Lu, Y.; Mei, Q.; Zhai, C. Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA. *Inf. Retr.* **2011**, *14*, 178–203. [[CrossRef](#)]
59. Thomas, H. Probabilistic latent semantic analysis. *arXiv* **2013**, arXiv:1301.6705.

60. Julian, M.; Leskovec, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013.
61. Wang, H.; Luo, N. Collaborative filtering enhanced by user free-text reviews topic modelling. In Proceedings of the 2014 International Conference on Information and Communications Technologies, Nanjing, China, 15–17 May 2014.
62. Rodrigo, M.; Valiati, J.F.; Neto, W.P.G. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Syst. Appl.* **2013**, *40*, 621–633.
63. Fouzia, S.S.; Hussain, A.R.; Hameed, M.A. Supervised opinion mining of social network data using a bag-of-words approach on the cloud. In *Proceedings of the Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Gwalior, India, 4 December 2012; Springer: New Delhi, India, 2013.
64. Kyoungok, K.; Lee, J. Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction. *Pattern Recognit.* **2014**, *47*, 758–768.
65. Chin, C.C.; Chen, Z.; Wu, C. An unsupervised approach for person name bipolarization using principal component analysis. *IEEE Trans. Knowl. Data Eng.* **2011**, *24*, 1963–1976. [[CrossRef](#)]
66. Poirier, D.; Fessant, F.; Tellier, I. Reducing the cold-start problem in content recommendation through opinion classification. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'10), Toronto, ON, Canada, 31 August–3 September 2010; Volume 1, pp. 204–207.
67. Zhang, W.; Ding, G.; Chen, L.; Li, C.; Zhang, C. Generating virtual ratings from chinese reviews to augment online recommendations. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 1–17. [[CrossRef](#)]
68. Shen, R.P.; Zhang, H.R.; Yu, H.; Min, F. Sentiment based matrix factorization with reliability for recommendation. *Expert Syst. Appl.* **2019**, *135*, 249–258. [[CrossRef](#)]
69. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004.
70. Nozomi, K.; Inui, K.; Matsumoto, Y. Extracting aspect-evaluation and aspect-of relations in opinion mining. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 1065–1074.
71. Ana-Maria, P.; Etzioni, O. Extracting product features and opinions from reviews. In *Natural Language Processing and Text Mining*; Springer: London, UK, 2007; pp. 9–28.
72. Khan, K.; Baharudin, B.; Khan, A.; Ullah, A. Mining opinion components from unstructured reviews: A review. *J. King Saud Univ. Comput. Inf. Sci.* **2014**, *26*, 258–275. [[CrossRef](#)]
73. Qi, L.; Li, C. Comparison of model-based learning methods for feature-level opinion mining. In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Lyon, France, 22–27 August 2011.
74. Li, F.; Han, C.; Huang, M.; Zhu, X.; Xia, Y.J.; Zhang, S.; Yu, H. Structure-aware review mining and summarization. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010.
75. Samaneh, M.; Ester, M. Opinion digger: An unsupervised opinion miner from unstructured product reviews. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010.
76. Gayatree, G.; Kakodkar, Y.; Marian, A. Improving the quality of predictions using textual information in online user reviews. *Inf. Syst.* **2013**, *38*, 1–15.
77. 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.
78. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February, 2017.
79. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018.

80. Wu, L.; Quan, C.; Li, C.; Wang, Q.; Zheng, B.; Luo, X. A context-aware user-item representation learning for item recommendation. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–29. [[CrossRef](#)]
81. Liu, H.; Wang, Y.; Peng, Q.; Wu, F.; Gan, L.; Pan, L.; Jiao, P. Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing* **2020**, *374*, 77–85. [[CrossRef](#)]
82. Cao, J.; Hu, H.; Luo, T.; Wang, J.; Huang, M.; Wang, K.; Wu, Z.; Zhang, X. Distributed design and implementation of svd++ algorithm for e-commerce personalized recommender system. In *Proceedings of the 3th National Conference on Embedded System Technology, Beijing, China, 10–11 October 2015*; Springer: Singapore, 2015.
83. Andriy, M.; Salakhutdinov, R.R. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2007.
84. Chong, W.; Blei, D.M. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011*.
85. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, The MIT Press: Cambridge, MA, USA, 2007.
86. Tan, Y.; Zhang, M.; Liu, Y.; Ma, S. *Rating-Boosted LATENT Topics: Understanding Users and Items with Ratings and Reviews*; IJCAI: Vienna, Austria, 2016; Volume 16.
87. 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*.
88. Bao, Y.; Fang, H.; Zhang, J. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014*.
89. Cheng, Z.; Ding, Y.; Zhu, L.; Kankanhalli, M. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018*.
90. Chin, J.Y.; Zhao, K.; Joty, S.; Cong, G. ANR: Aspect-based neural recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018*.
91. Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; Ma, S. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, Gold Coast, QLD, Australia, 6–11 July 2014*.
92. Musto, C.; de Gemmis, M.; Semeraro, G.; Lops, P. A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017*.
93. Da' u, A.; Salim, N.; Rabi, I.; Osman, A. Weighted aspect-based opinion mining using deep learning for recommender system. *Expert Syst. Appl.* **2020**, *140*, 112871.
94. Caputo, A.; Basile, P.; de Gemmis, M.; Lops, P.; Semeraro, G.; Rossiello, G. SABRE: A sentiment aspect-based retrieval engine. In *Information Filtering and Retrieval*; Springer: Cham, Switzerland, 2017; pp. 63–78.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).