

## Article

# Applying Deep Learning Models to Analyze Users' Aspects, Sentiment, and Semantic Features for Product Recommendation

Chin-Hui Lai \*  and Kuo-Chiuan Tseng

Department of Information Management, Chung Yuan Christian University, Taoyuan City 32023, Taiwan;  
jim5260tw@gmail.com

\* Correspondence: chlai@cycu.edu.tw

**Abstract:** As there is a huge amount of information on the Internet, people have difficulty in sorting through it to find the required information; thus, the information overload problem becomes a significant issue for users and online businesses. To resolve this problem, many researchers and applications have proposed recommender systems, which apply user-based collaborative filtering, meaning it only considers the users' rating history to analyze their preferences. However, users' text data may contain users' preferences or sentiment information, and such information can be used to analyze users' preferences more precisely. This work proposes a method called the aspect-based deep learning rating prediction method (ADLRP), which can extract the aspects, sentiment, and semantic features from users' and items' reviews. Then, the deep learning method is used to generate users' and items' latent factors. According to these three features, the matrix factorization method is applied to make rating predictions for items. The experimental results show that the proposed method performs better than the traditional rating prediction methods and conventional artificial neural networks. The proposed method can precisely and efficiently extract the sentiments and semantics of each aspect from review texts and enhance the prediction performance of rating predictions.



**Citation:** Lai, C.-H.; Tseng, K.-C. Applying Deep Learning Models to Analyze Users' Aspects, Sentiment, and Semantic Features for Product Recommendation. *Appl. Sci.* **2022**, *12*, 2118. <https://doi.org/10.3390/app12042118>

Academic Editors: Teen-Hang Meen and Chun-Yen Chang

Received: 15 January 2022

Accepted: 14 February 2022

Published: 17 February 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/>).

**Keywords:** recommender system; text mining; opinion mining; topic modeling; attention mechanism; deep learning; CNN; aspect detection

## 1. Introduction

Due to the development of Web 2.0 and the popularity of mobile devices, everyone is free to share their opinions and publish articles on the Internet. As a result, the amount of information on the Internet grows exponentially with time; as a further result, users need to spend a lot of time to find the right data that meet their needs; thus, the efficiency of decision making is reduced. In addition, compiling and analyzing a huge amount of data has become a major problem. To solve the above-mentioned problems, many scholars have proposed recommendation methods or systems to filter unnecessary information. Recommender systems mainly analyze users' preferences, behaviors, and community relations, prior to recommending items that may be of interest in future searches [1–4]. Traditional recommendation methods analyze user preferences based on ratings; however, ratings cannot always accurately represent users' preferences.

Many e-commerce and social networking sites, such as Amazon.com and Yelp.com, allow users to comment on products, give ratings, or engage in discussions. The text review data may contain a lot of potentially important information. Different aspects, sentiment, and semantics hidden in the reviews can express user preferences and product features. As the review data are unstructured, it is difficult to consider and acquire lexical or grammatical issues for analysis. For analyzing text reviews, some scholars proposed methods such as topic models [5,6] and machine learning methods [7,8] to automatically extract the contexts of articles or identify valuable knowledge. These studies focused on determining how to generate the above-mentioned aspects, their polarities, or sentiments from user reviews.

However, these methods only consider these three factors simultaneously when extracting the aspects from the review texts.

Additionally, the semantics and sentiments which are implicit in text reviews can reveal users' preferences. For extracting the semantics from the texts, deep learning methods can be used to analyze users and reviews on products; then, collaborative filtering can be used to predict ratings [9]. Zheng et al. [10] used deep cooperative neural networks to predict ratings based on user behavior and product reviews. To understand the expression of sentiments in the text, analyzing the emotion-related vocabulary contained in texts can provide insights into the sentiments and attitudes conveyed by users in their texts [11]. On the other hand, sentiment polarity or intensity can be used to represent the users' emotional performance [12,13]. These studies show that deriving semantics or sentiments from reviews or text can capture user preferences and perceived product qualities. Using deep learning or neural network methods can improve the performance and accuracy of both prediction and classification. However, these studies only analyzed the semantics or sentiments in the reviews, without considering user-preferred aspects implied in user reviews. To improve the accuracy of recommendation methods, determining how to extract accurate user preferences and features based on user behavior is an important issue. Besides users' rating data, developing the methods for identifying user-preference aspects, semantic and sentiment features hidden in text reviews, and combining the implicit and explicit information in a recommendation method help to generate insights into user preferences and product features.

To address the problem in the existing studies, this work proposes an aspect-based deep learning rating prediction method (ADLRP) for review websites, which consists of four main components: aspect detection, sentiment analysis, semantic analysis, and rating prediction. In the aspect of detection, sentence-level latent Dirichlet allocation (SLDA) [6,14] was used to analyze the topics at the sentence level, in order to identify the implicit aspect of the reviews. The word embedding algorithm was then used to generate aspect embedding vectors. Sentiment analysis calculates the sentiment intensity of each aspect of the reviews and builds the sentiment vector with the neural network method. The semantic analysis uses hierarchical attention networks (HAN) [15] to calculate word and sentence attention in user and product reviews. The user and product aspect semantic vectors are generated based on the weights generated. In this study, user latent factors and product latent factors were generated by integrating the aspect, semantic, and sentiment vectors of users and products, and then by adopting the convolutional neural networks (CNN) method. Finally, matrix factorization (MF) [16] was leveraged to predict products that might be of interest to users in the future. The experimental results show that this study features the best prediction accuracy by using aspect, aspect sentiment, and aspect semantic features for rating prediction. Additionally, applying the deep learning method in making recommendations also leads to better performance and accuracy compared to other methods.

In summary, the main contributions of this work include the following.

- We propose a novel ADLRP method herein, which could effectively extract the aspect, sentiment, and semantic features from text reviews and combine these features with ratings for rating prediction. Our method integrates both implicit and explicit information to analyze user preferences and product features, and thus achieves better predictive performance.
- A blend of features can achieve more accurate user preferences and product evaluation features, thereby enabling merchants to better understand users' preferences and provide them with more accurate product recommendations.
- The deep learning methods and attention mechanism used in our method can effectively extract and train the user and product features to improve the accuracy of rating prediction.

This paper comprises five sections. Section 2 explores the literature on deep learning, semantic and sentiment analysis, and matrix factorization. Section 3 introduces the proposed

method, i.e., ADLRP. Section 4 presents the experimental results and discussion of this study and compares it with other related research methods. Section 5 concludes the study.

## 2. Related Works

### 2.1. Deep Learning Methods

In natural language processing, deep learning methods are used for sentiment analysis and opinion mining. In this section, we discuss the common deep learning methods.

#### 2.1.1. Convolutional Neural Network

The Convolutional Neural Network (CNN) has been applied to improve the accuracy of classification and prediction in natural language processing, such as text classification, sentiment analysis, language modeling, and information retrieval [9]. Li, Li, Lee, and Kim [8] proposed an innovative deep learning architecture that considers prediction errors (reliability) to enhance the quality of predictions and recommendations. CNN consists of a hidden layer, a convolutional layer, and a pooling layer. In natural language processing, word embedding and CNN can be appropriately combined to effectively extract the semantic features of texts [17].

In CNN, text data are filtered to generate features through the convolutional layer; the feature matrix is then reduced in dimensionality and representative features are extracted by the pooling layer. Finally, all the features are concatenated in a fully connected layer and the results are output [18]. Wang et al. [19] suggested that the sentences in an article can represent its region features, and applied regional convolution neural networks (RCN) employed to analyze the sentences in articles. The long short-term memory (LSTM) model is used to cascade regional features to find the relationships between regions (i.e., sentences) in an article to improve the accuracy of article sentiment classification. Zheng, Noroozi, and Yu [10] integrated deep learning and collaborative filtering methods to generate user latent factors and product latent factors, respectively, with a CNN for rating prediction; the results showed that CNN can improve prediction accuracy.

#### 2.1.2. Recurrent Neural Network

A Recurrent Neural Network (RNN) is a neural network that transmits back the previous output value, records the theme in the neurons, and considers the previous output results in the following calculation of the output results, signifying that the neural network has the ability to remember. RNN performs very well in sequence data, language analysis, word prediction, and recognition [20]. However, when the data are trained with Long Sequences, the gradient will grow or decline, which can also cause the vanishing gradient problem or exploding gradient problem in RNN [21]. To solve this problem, some scholars have modified the architecture of RNN and proposed some improved models, such as long short-term memory (LSTM) and gated recurrent unit (GRU) [22]. The LSTM method adds a forget gate to the RNN to filter unnecessary information, while GRU adds an Update Gate and Reset Gate to the RNN to select the required information and discard the redundant information.

In natural language processing, considering the contextual relationship can improve the accuracy of prediction and classification. Bidirectional Recurrent Neural Networks (Bi-RNN), as proposed by scholars [15], considers the context of an article (from front to back and back to front) and encodes the article from two directions. Generating more short-term dependencies also improves the effectiveness of the model [23]. Bahdanau, Cho, and Bengio [22] improved the accuracy of machine translation by feeding articles into a bidirectional recurrent neural network in order to identify the contextual relationships in the articles. Tang et al. [24] used the CNN and LSTM methods to extract features and input them into the Gated Recurrent Neural Network to obtain the relationships between sentences in an article and enhance the accuracy of sentiment classification.

### 2.1.3. Attention Mechanism

In many studies on natural language processing, attention mechanisms [15,22,25] are used to generate the weights of sentences or words. Wang, Huang, and Zhao [25] used the attention mechanism in the LSTM model, whereby the influential words in an article were defined in seeking to extract the aspects of the article and improve the accuracy of article classification. Yang, Yang, Dyer, He, Smola, and Hovy [15] proposed the hierarchical attention network (HAN), based on the attention mechanism, to analyze the importance of words and sentences in an article by considering the contextual relationships in the article with Bi-RNN; this attention network was eventually used to classify the sentiment in articles. A review semantics-based model [26] can extract users' assessment semantics from reviews by using CNN and then model users' assessment actions for predicting ratings by using an attention network. Liu et al. [27] proposed an attention-based adaptive memory network (AAMN) that uses an attention mechanism to classify the importance levels of different reviews for modeling the adaptive features of users and items.

### 2.2. Semantic and Sentiment Analysis

As texts are unstructured data, we often convert text into numerical or vector forms for generating relevant computations. In general, the conversion of texts into semantic vectors can be divided into word vectors and sentence vectors. There are two types of word vector conversion: (1) One-hot representation and (2) Distributed representation. One-hot representation adopts 0 and 1 to represent a vector, where 1 represents the presence of a specific word. The number of document words represents the dimensionality of the vector. In the text vector of each word, only one dimension is 1, while most of the dimensions are 0. Word embedding is commonly used to convert words into word vectors with Distributed representation [24,28]. This method is trained from the corpus to build word vectors and preserve word-to-word relationships; the vectors are of a fixed dimension. Common approaches for word embedding are Word2Vec [29] and GloVe [30]. Word2Vec builds word vectors based on the continuous bag-of-words (CBOW) model and skip-gram. GloVe considers both local and global information in the lexicon and uses distributed statistics to build word vectors. These two algorithms have been applied in many studies to generate word vectors and perform sentiment analysis [7,9,19,31].

Sentence vectors can be generated by summing or averaging all word vectors in a sentence to represent the semantic vector of the sentence. Tang, Qin, and Liu [24] used three filters of different lengths (1, 2, and 3) in CNN to analyze the sentences in an article. The important features of the three filters, i.e., unigram, bigram, and trigram, are then extracted by max-pooling. Finally, these three features of different lengths are averaged for the sentence vector. Dang et al. [32] combined sentiment analysis and genre-based similarity in the collaborative filtering method; it uses Bidirectional Encoder Representation from Transformers (BERT) to extract genres and features from user reviews in the deep-learning-based sentiment analysis in order to improve the recommendation quality of streaming platforms.

The dictionary method is often used in sentiment analysis to analyze the sentiment features in the text. To quantify the sentiments implied by a text, many studies adopt sentiment dictionaries to identify the sentiment-related words in the text and calculate the sentiment intensity and polarity of the text. Commonly used emotion dictionaries include ANEW [14], SentiWordNet [33], SenticNet [34], and HowNet [35]. The higher the sentiment intensity score, the more positive the sentiment expressed in an article; conversely, the lower the sentiment intensity score, the more negative the sentiment expressed in the article [12]. Hamouda and Rohaim [36] applied SentiWordNet for sentiment analysis in product reviews and used averaging and summing to calculate the sentiment intensity scores of words. Their study found that averaging the sentiment word scores in an article derived better analysis results.

### 2.3. Matrix Factorization

Matrix factorization (MF), one of the most commonly used models in collaborative filtering methods, uses a latent factor model to analyze rating information and identify users as well as item characteristics [16,17]. The matrix factorization method factorizes the user rating matrix into two matrices: user latent factors and item latent factors; the inner product of the two matrices can help predict the rating of unrated items. Equation (1) represents the predicted rating of user  $u$  for item  $i$  (i.e.,  $\hat{r}_{u,i}$ ):

$$\hat{r}_{u,i} = x_i^T \cdot y_u \quad (1)$$

where  $x_i$  and  $y_u$  are the item latent factors and user latent factors, respectively. The ultimate goal of matrix factorization is to minimize the objective function-least square error, and find the best solution, as expressed in Equation (2).

$$\min_{x^*, y} \sum_{(u,i) \in K} \left( r_{u,i} - x_i^T y_u \right)^2 + \lambda \left( \|x_i\|^2 + \|y_u\|^2 \right) \quad (2)$$

$K$  is the set of implicit preferences of user  $u$  for product item  $i$ ,  $r_{u,i}$  represents the rating value of the training data, and  $r_{u,i} - x_i^T y_u$  presents the error between the actual rating and the predicted rating. To avoid overfitting, a constant  $\lambda$  is added to control the regularization. Matrix factorization models often use learning algorithms, such as Stochastic Gradient Descent (SGD) and Alternating Least Square (ALS), to tune models [16] and achieve the best prediction results.

Users' text reviews can also be leveraged to analyze their preferences. Baltrunas et al. [37] coded user and product characteristics according to the contextual information of an article; they then modified the traditional matrix factorization model for rating prediction. Seo, Huang, Yang, and Liu [9] applied the attention mechanism to a CNN to analyze the reviews of users and products and establish the latent factors of users and products, respectively. The matrix factorization method was then used to improve the accuracy of rating predictions. Zheng, Noroozi, and Yu [10] used CNN to analyze the word vectors in reviews and generate the vectors of latent factors for users and products, respectively; they then input the two feature vectors into the matrix factorization method to predict the ratings. The differentiating users' preferences and items' attractiveness (DUPA) method uses a hybrid probabilistic matrix factorization model to predict user ratings based on diverse users' preferences extracted from auxiliary information [38]; it also uses attention in CNN to model products' textual attractiveness to different users in different perspectives of products.

The latent factor model uses a user-item rating matrix to make predictions; however, such a matrix is commonly high-dimensional and sparse (HiDS). Wu et al. [39] proposed an  $L_1$ -and- $L_2$ -norm oriented latent factor ( $L^3F$ ) model, which aggregates  $L_1$  norm-oriented Loss's robustness and  $L_2$  norm-oriented Loss's stability; it well describes an HiDS rating matrix, and has high prediction accuracy, despite missing data in the matrix. The SGD-based latent factor analysis (LFA) model cannot handle a large-scale HiDS matrix due to the difficulty in tuning its learning rate. A P<sup>2</sup>SO-based LFA (PLFA) model [40] was proposed to efficiently construct a learning rate swarm in an SGD-based LFA model without any accuracy loss or computation burden. It also precisely presents an HiDS matrix, improves the prediction accuracy, and enhances computation efficiency. The latent factor model can also be used for quality-of-service (QoS) predictions. For example, a posterior-neighborhood-regularized latent factor (PLF) model constructs neighborhoods based on full information in a user-service matrix and has high accuracy of QoS prediction [41]. A data-characteristic-aware latent factor (DCALF) model can present a user-service QoS matrix to model the user/service neighborhoods without any information loss, as well as to detect noise from QoS data [42].

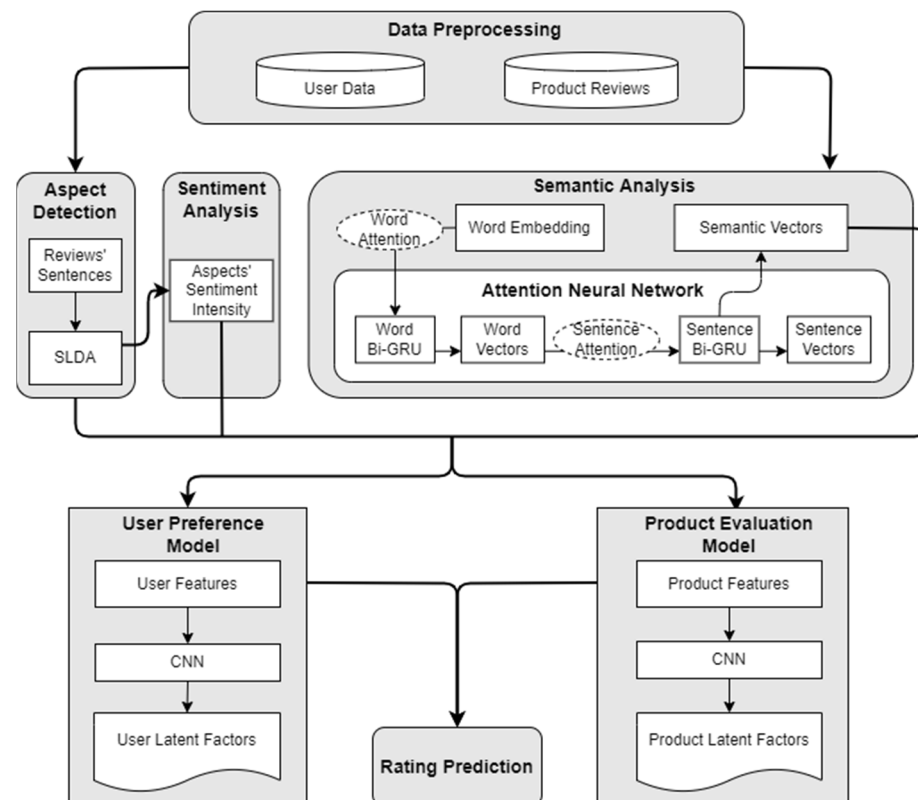


### 3. The Proposed Method

#### 3.1. Research Overview

This study proposes an aspect-based deep learning rating prediction method (ADLRP) for review sites. This method is used to analyze the aspects of user reviews and their preferences for different aspects, as well as the product reviews according to different aspects. The user and product features are then integrated to predict future products of interest for users. Previous studies have usually only analyzed the historic rating values for predictions and recommendations, but ignored users' preferences and product features implied in the text reviews. In addition, different aspects, sentiment, and semantics hidden in the text reviews can express user preferences or product evaluation. If we only analyze user and product ratings, we cannot thoroughly understand the user's preferences and needs. Therefore, considering user-preference aspects, semantic features, and sentiment features hidden in text reviews helps to gain insight into user preferences and product features. Combining the implicit (i.e., the aspect, semantic, and sentiment features) and explicit information (i.e., user ratings) in a rating prediction method can accurately and effectively recommend products that may be of interest to users.

Figure 1 shows the research process of the proposed method. This study takes the Electronics category of Amazon Datasets (<https://jmcauley.ucsd.edu/data/amazon/>, accessed on 31 July 2020) as the dataset for the experiment. This dataset contains user information, product information, ratings, reviews, and other data. The method proposed herein is divided into seven modules: data pre-processing module, aspect detection module, sentiment analysis module, semantic analysis module, user preference model, product evaluation model, and rating prediction module.



**Figure 1.** The overview of the proposed method.

First, we set the data filtering criteria for the product reviews of users to select the eligible users and reviews, and then conducted pre-processing. Regarding the aspect detection module, the aspects of the sentences in the reviews were analyzed to infer the wider aspects of the article and then build the aspect embedding vector. Sentiment analysis

was used to calculate the sentiment intensity in each aspect of the reviews. The sentiment feature vector of the aspect was then built via the neural network method. The semantic analysis uses a hierarchical attention network (HAN) to calculate the weights of words and sentences in user and product reviews. The resulting weights are used to generate the aspect semantic vectors of the users and products. In the user preference model, the aspect vectors, aspect sentiment scores, and aspect semantic vectors, as extracted from user reviews, are integrated and fed into a user-based CNN to generate the user latent factors. Similarly, the product evaluation model generates product latent factors in the same way as the user preference model. Finally, the matrix factorization (MF) method was used to integrate the latent factors of users and products to predict the future products that users may be interested in.

### 3.2. Aspect Detection

Analyzing the user reviews and the product reviews in different aspects can reveal the users' preferences for different aspects. To analyze the implied aspects of such reviews, this study first decomposed the textual content of reviews into single sentences and then used the topic model, i.e., sentence-level latent Dirichlet allocation (SLDA) [6] for aspect detection. SLDA infers the topic (i.e., aspect) of a sentence based on the words contained in each sentence. Each sentence belongs to only one aspect, and an aspect is composed of multiple feature words. We used the word embedding algorithm, GloVe [30], to convert the feature words contained in an aspect into a semantic vector. Finally, the semantic vectors of the same aspect derived from users and product reviews were summed up, respectively, to generate the aspect embedding vectors of users and products.

Assuming that user  $u$ 's reviews contain  $m$  aspects, and each aspect contains  $n$  feature words, based on the result of GloVe, the feature words  $k$  contained in the aspects are converted into a word vector  $\vec{f}_{u,x}^k$ . The word vectors of the same aspect are then summed to represent the vector of  $x$ .  $A_{u,x}$  denotes the aspect  $x$  vectors of user  $u$ , and the formula is shown as follows:

$$A_{u,x} = \sum_{k=1}^n \vec{f}_{u,x}^k \quad (3)$$

Assume that there are  $m$  aspects in the product  $p$  reviews, and that each aspect contains  $n$  feature words; similarly, according to the results of GloVe, feature words contained in aspect  $l$  are converted into a word vector  $\vec{f}_{p,l}^k$ . The word vectors of the same aspect are summed to obtain the aspect vector  $A_{p,l}$ ; the formula is shown as follows:

$$A_{p,l} = \sum_{k=1}^n \vec{f}_{p,l}^k \quad (4)$$

### 3.3. Sentiment Analysis

#### 3.3.1. Aspect Sentiment Intensity

In this study, the sentiment dictionary SentiWordNet [33] was used to calculate the sentiment scores of the sentiment-featured words in each sentence of the user reviews. The sentiment scores of the words in each sentence are summed to obtain the sentiment intensity of a single sentence. In the user reviews, the sentiment intensity of sentences in the same aspect is summed to generate the sentiment intensity of the aspect. Similarly, the sentiment intensity of the product aspect is generated via the same method. The higher the sentiment intensity of the aspect, the more the user values the aspect.

Assume that user  $u$  contains  $a$  sentences in aspect  $x$ , and that each sentence contains  $m$  sentiment-featured words. We used the sentiment dictionary SentiWordNet to obtain the scores of the sentiment-featured words contained in each sentence  $k$ ; then, the scores were summed to obtain the sentiment intensity (i.e.,  $o_{u,x}^k$ ). We then summed the sentiment

intensity of  $a$  sentences, as contained in the user's aspect  $x$ , to obtain the sentiment intensity  $US_{u,x}$ , as shown in Equation (5):

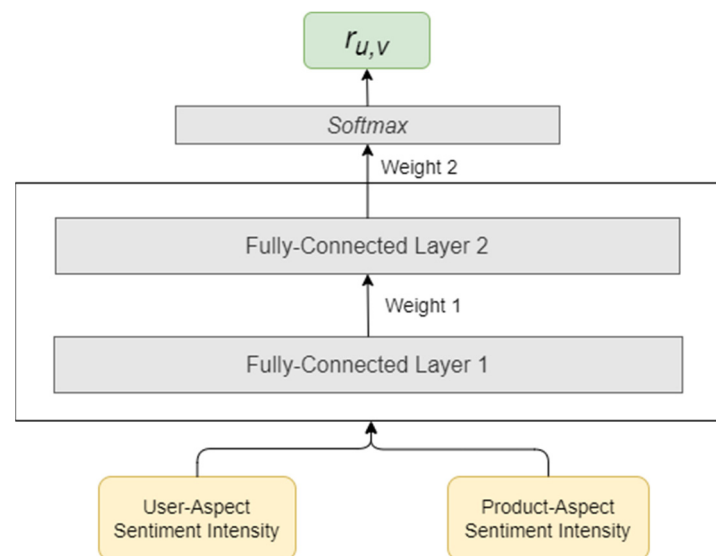
$$US_{u,x} = \sum_{k=1}^a o_{u,k}^x \quad (5)$$

We used the same method to generate the sentiment scores of product aspects. Assume that product  $p$  contains  $b$  sentences in aspect  $l$ , and that each sentence contains  $m$  sentiment-featured words. We used SentiWordNet to obtain the scores of the sentiment-featured words contained in each sentence  $k$ , and then summed the sentiment scores to obtain the sentiment intensity. We then summed the sentiment intensity of  $b$  sentences, as contained in product aspect  $l$ , to obtain the sentiment intensity  $PS_{p,l}$  of product aspect  $p$ , as shown below:

$$PS_{p,l} = \sum_{k=1}^b o_{p,l}^k \quad (6)$$

### 3.3.2. Aspect Sentiment Vector

As aspect sentiment intensity is purely numerical, to combine it with other feature vectors, this study adopted an artificial neural network to convert the numerical values of aspect sentiment intensity into feature vectors. Based on the study of Zhang et al. [43], an artificial neural network was employed by this research to build a rating prediction model, as shown in Figure 2. This artificial neural network architecture consists of one input layer, two fully connected layers, and one output layer. The input layer is the user-product aspect sentiment intensity. The fully connected layer is used to concatenate the user's sentiment intensity characteristics for the product. The output layer uses a *Softmax* function to classify the output into five categories, which represent the predicted ratings of the reviews (5-star rating). Finally, the user and product weights are extracted from the model to convert aspect sentiment intensity into an aspect sentiment vector with a fixed dimension.



**Figure 2.** Generating the aspect sentiment vector.

Assume that  $r_u$  is the predicted user rating,  $r_i$  is the predicted product rating,  $g$  is a nonlinear function,  $w_u$  is the weight of the user model,  $w_i$  is the weight of the product model, and  $b_u$  and  $b_i$  are the bias of the user model and the product model, respectively. The prediction scores are calculated, as follows:

$$r_u = g(US_{u,x} \times w_u + b_u) \quad (7)$$

$$r_i = g(PS_{p,l} \times w_i + b_i) \quad (8)$$



The user and product weights are extracted from the model and unemployed to present the aspect sentiment vector. Assume that  $\vec{AS}_{u,x}$  is the sentiment vector of user  $u$  in aspect  $x$  and  $\vec{AS}_{p,l}$  is the sentiment vector of product  $p$  in aspect  $l$ . These two sentiment vectors are presented by Equations (9) and (10), respectively. As their vector dimensions are the same as the aspect vector (Section 3.2) and the semantic vector (Section 3.4), integration can be subsequently conducted:

$$\vec{AS}_{u,x} = U_{u,x} \times w_u \quad (9)$$

$$\vec{AS}_{p,l} = PS_{p,l} \times w_i \quad (10)$$

### 3.4. Semantic Analysis

In product reviews, users express their sentiments, feelings, preferences, etc., in words. To explore the implicit content of user-product reviews, many studies have used the attention mechanism to analyze the semantic features in sentences [9,15,44]. There is a hierarchical relationship between the words and sentences of a review. Multiple words form a sentence, and multiple sentences can be organized into one review. Based on the hierarchical relationships in the text, this study used a hierarchical attention neural network [15] to analyze the semantic meanings in reviews.

This study uses GloVe [30], a word embedding method, to convert the texts of user and product reviews into word vectors. The word vectors were then fed into a hierarchical attentional neural network for analysis, as shown in Figure 3. This method calculates the word vectors and word weights, as well as the sentence vectors and sentence weights, based on the hierarchical and contextual relationships in the review text. Finally, the sentence vector and the sentence weight are multiplied to produce a weighted sentence vector. Similar to the user and product aspects, the weighted sentence vectors of the same aspect were summed to generate the aspect semantic vectors of users and products, respectively.

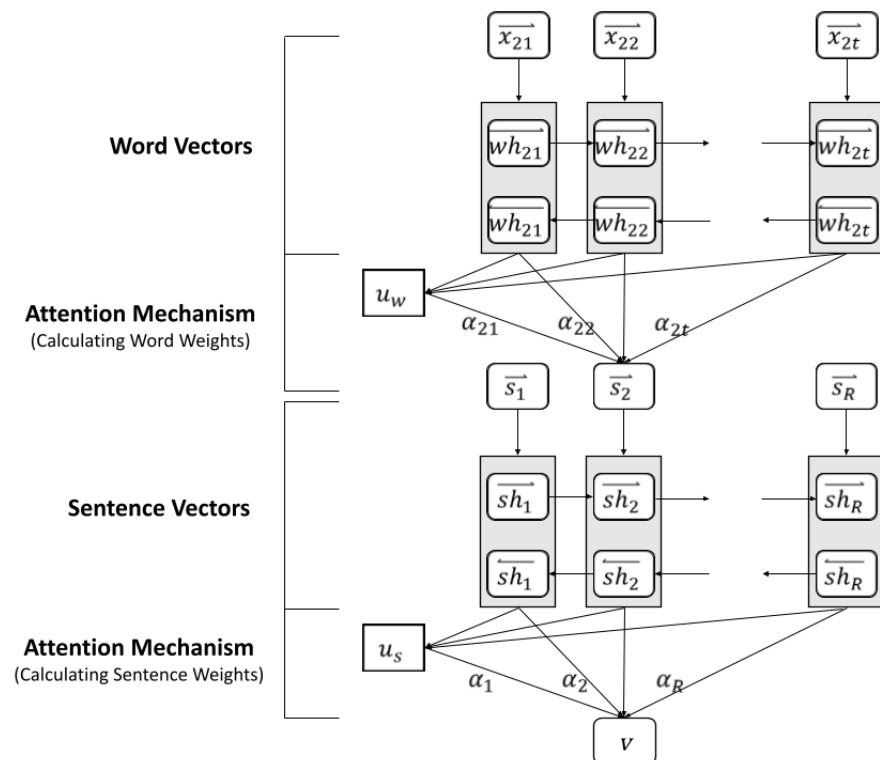


Figure 3. Attention Neural Network.

Assume that the user reviews contain  $m$  aspects and that each aspect contains  $n$  sentences. According to the analysis results of the hierarchical attention neural network, we multiplied the weight  $\alpha_{u,x}^i$  of sentence  $i$  in aspect  $x$  by vector  $\vec{sh}_{u,x}^l$  to obtain the weighted sentence vector. Then, all the sentence vectors in aspect  $x$  were summed to obtain the semantic vector  $\vec{V}_{u,x}$  of user aspect  $x$ , as shown in the following equation:

$$\vec{V}_{u,x} = \sum_{x=1}^m \sum_{i=1}^n \alpha_{u,x}^i \cdot \vec{sh}_{u,x}^l \quad (11)$$

Like the user aspect semantic method, assume that the product reviews contain  $m$  aspects and that each aspect contains  $n$  sentences. According to the results of the hierarchical attention neural network, we multiplied sentence weight  $\alpha_{p,l}^i$  contained in aspect  $l$  by sentence vector  $\vec{sh}_{p,l}^l$  to obtain the weighted sentence vector. All sentence vectors in aspect  $l$  were then summed to obtain the semantic vector  $\vec{V}_{p,l}$  of user aspect  $l$ , as shown in the following equation.

$$\vec{V}_{p,l} = \sum_{l=1}^m \sum_{i=1}^n \alpha_{p,l}^i \cdot \vec{sh}_{p,l}^l \quad (12)$$

### 3.5. User Preference Model

In natural language processing in the past few years, the CNN method has often been used for analysis [9,10,19]. CNN uses filters to extract features from the input data, selects important features by the pooling method, and finally integrates all the filtered results via the fully connected layer. As the CNN method can effectively extract regional features and improve the performance of computing, we used CNN architecture in this study to extract user features in the user preference model.

The user preference model assumes that user reviews contain  $m$  aspects. Regarding aspect  $x$  in the user reviews, we integrated aspect vector  $\vec{A}_{u,x}$  (Section 3.2), aspect sentiment vector  $\vec{AS}_{u,x}$  (Section 3.3.2), and aspect semantic vector  $\vec{V}_{u,x}$  (Section 3.4), as obtained in the previous steps, as shown in Equation (13). The common feature integration methods include concatenation and summation [43]. The concatenation method naturally integrates feature vectors in a concatenated manner, which increases the dimensionality of the vectors. The summation method adds up the content of the corresponding dimensions in the vectors; the dimensionality of the vectors remains unchanged after summation. This study used the summation method to integrate the feature vectors of three users:

$$\vec{UP}_{u,x} = \vec{A}_{u,x} + \vec{AS}_{u,x} + \vec{V}_{u,x} \quad (13)$$

As different aspects are contained in the reviews of users, we expressed the feature vector of user  $u$  as  $\vec{UP}_u = \{(\vec{A}_{u,1} + \vec{AS}_{u,1} + \vec{V}_{u,1}), \dots, (\vec{A}_{u,m} + \vec{AS}_{u,m} + \vec{V}_{u,m})\}$ . This feature vector was then fed into CNN to train the user features. The CNN architecture used in this study consists of one input layer, two convolutional layers, one pooling layer, and one fully connected layer. Through convolutional calculation, user latent factor  $U$  was finally obtained.

### 3.6. Product Evaluation Model

Similar to the user preference model, we used the same approach to build a product evaluation model to analyze individual product reviews under different aspects. The method described in Section 3.5 was used to characterize the product aspects. Regarding aspect  $l$  in the product reviews, we used the summation method to integrate the product aspect vector  $\vec{A}_{p,l}$ , aspect sentiment vector  $\vec{AS}_{p,l}$ , and aspect semantic vector  $\vec{V}_{p,l}$ , as obtained in the previous steps, as shown in Equation (14):

$$\vec{PA}_{p,l} = \vec{A}_{p,l} + \vec{AS}_{p,l} + \vec{V}_{p,l} \quad (14)$$

As different aspects are contained in the reviews of products, we input the review feature vector of product  $p$   $\vec{PA}_p = \{(\vec{A}_{p,l} + \vec{AS}_{p,l} + \vec{V}_{p,l}), \dots, (\vec{A}_{p,m} + \vec{AS}_{p,m} + \vec{V}_{p,m})\}$  into CNN for training, in order to obtain product latent factor  $Q$ .

### 3.7. Rating Prediction

In a recommender system, the matrix factorization (MF) technique factorizes the user rating matrix into a user feature matrix and a product feature matrix, which are mapped into vector space  $\mathbb{R}^K$ . The matrix inner product calculation is then used to predict the users' rating of the unrated product items. Based on the MF method [16], the user latent factor  $U$  and product latent factor  $P$ , as generated from the user preference model (Section 3.5) and product evaluation model (Section 3.6), respectively, were input into the MF method to produce the predicted ratings. However, machine learning methods (e.g., CNN) require several iterations to adjust the model parameters to achieve the best prediction results. Therefore, to obtain better rating prediction results in this study, an additional layer of MF was added to the deep learning model in Sections 3.5 and 3.6 to predict the products that users may be interested in and the rating value of products. The calculation of the predicted rating is shown in Equation (15):

$$r_{u,p}^{\hat{}} = Q_p^T \cdot U_u \quad (15)$$

where  $r_{u,p}^{\hat{}}$  is the predicted score;  $U_u$ , the user  $u$ 's latent factor; and  $Q_p$ , the product  $p$ 's latent factor. To minimize the difference between the original and predicted ratings, this study used stochastic gradient descent (SGD) to minimize the objective function (i.e., Equation (16)) and establish the most suitable user latent factors and product latent factors. The objective of the SGD method is to identify the optimal regional solution of the function and randomly select samples to update the parameters and converge to the global minimum after several iterations in order to avoid generating regional minima:

$$\min_{Q^* U^*} \sum_{u,p} \left( r_{u,p} - Q_p^T \cdot U_u \right)^2 + \lambda_Q \|Q_p\|^2 + \lambda_U \|U_u\|^2 \quad (16)$$

where  $r_{u,p}$  is the actual rating;  $\lambda_Q$  and  $\lambda_U$  are regularization parameters used to avoid overfitting;  $Q_p$  is product  $p$ 's latent factor, and  $U_u$  is user  $u$ 's latent factor. The parameters in Equation (16) were updated according to the two rules [45,46] defined in Equations (17) and (18):

$$Q_p \leftarrow Q_p + \alpha (e_{u,p} U_u - \lambda_Q Q_p) \quad (17)$$

$$U_u \leftarrow U_u + \alpha (e_{u,p} Q_p - \lambda_U U_u) \quad (18)$$

where  $e_{u,p} = r_{u,p} - Q_p^T \cdot U_u$  presents the difference between the predicted rating and actual rating (i.e., the loss or error); and  $\alpha$  is the learning rate that can be selected differently for each factor matrix.

## 4. Experiment Evaluation

### 4.1. Data Collection

This study filtered the data of 2013 and 2014 from the Electronics category of Amazon Datasets (<https://jmcauley.ucsd.edu/data/amazon/>, accessed on 31 July 2020) for use as the dataset for the experiment. When analyzing the contents of user reviews, if the number of text reviews by users is too small to accurately analyze user preferences, sentiments, and semantics according to different aspects, it may affect the accuracy of rating prediction. Therefore, users with more than nine reviews and ratings, and products with more than five reviews, were selected for analysis. The filtered dataset consisted of 3409 users and 3440 products, with 46,279 reviews and ratings.

To conduct the experiment evaluation, 75% of the dataset was employed as a training set to train the feature vectors of the users and products, and then build a rating prediction model. In total, 5% of the dataset was used as a validation set to train the parameters of the proposed methods. The remaining 25% of the dataset was used as a testing set to verify the accuracy of the rating predictions. In data pre-processing, text pre-processing, such as word breaking, sentence breaking, useless word removal, punctuation removal, and stemming are performed in review articles. In aspect detection, we set each aspect as containing 30 feature words. The feature word vector is a pre-trained GloVe word vector [30] with 50 dimensions. In addition, the dimensionality of the aspect sentiment vector and aspect semantic vector were both set as 50 dimensions.

#### 4.2. Evaluation Indicators

To evaluate the prediction accuracy of the proposed methods, the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [3,47,48], which are predictive accuracy metrics widely used in the research field of recommender systems, were used as measurements of the rating prediction methods. MAE measures the average of the absolute difference between the predicted ratings and the true ratings. RMSE squares the error before summing it, and its result emphasizes large errors. The smaller the MAE and RMSE values, the more accurate the prediction, and vice versa. These two metrics are simple and easy to understand [49]. The MAE and RMSE are calculated as follows:

$$\text{MAE} = \frac{\sum_{r \in R} |TRS_r - RS_r|}{|R|} \quad (19)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{r \in R} (TRS_r - RS_r)^2}{|R|}} \quad (20)$$

where  $r$  is a review in the review set  $R$ ,  $TRS_r$  is a true rating score of review  $r$ , and  $RS_r$  is a predicted rating of review  $r$ .

#### 4.3. Experimental Results

This study conducted the experiments to set the parameters of the proposed methods and compared them with other related methods using different characteristics. The following subsection elaborates and discusses the results of each experiment in detail.

##### 4.3.1. Explanation of Experimental Methods

To validate the method proposed in this study, comparisons between our methods with other rating prediction methods were conducted. The following is a brief description of the relevant research methods.

1. Aspect-based Deep Learning Rating Prediction (ADLRP): The method proposed in this study. User and product reviews are analyzed, and aspect, aspect sentiment, aspect semantics, and other feature vectors are created. Through the training of the CNN, user latent factors and product latent factors are generated, and then matrix factorization is applied to predict ratings.
2. Deep Collaborative Neural Networks-CNN (DeepCoNN-CNN): DeepCoNN-CNN [10] analyzes user reviews and product reviews, and then uses a CNN to generate user latent factors and product latent factors. Finally, the matrix factorization method is used to predict the ratings.
3. Deep Collaborative Neural Networks-DNN (DeepCoNN-DNN): Similar to DeepCoNN-CNN, this method uses multilayer perceptron (MLP) to replace the CNN in DeepCoNN-CNN. This multilayer perceptron consists of two input layers (user and product reviews are input, respectively), three hidden layers, two fully connected layers (user latent factors and product latent factors are generated, respectively), and a final matrix decomposition layer for predicting ratings.

4. AutoRec: AutoRec is based on collaborative filtering and Auto-Encoder to infer user ratings for unrated products, as based on the user-item matrix [48]. Auto-Encoder is an unsupervised learning algorithm and a three-layer artificial neural network with an input layer, a hidden layer, and an output layer.
5. K-Nearest Neighbor (KNN): Similar to KNN-Mean, this method uses the ratings of  $K$  similar users to predict the target users' ratings [3,50].
6. K-Nearest Neighbor with Mean (KNN-Mean): Based on the collaborative filtering method, this method uses the ratings of  $K$  neighbors (i.e., similar users) and the average ratings to predict the target users' ratings for unrated products [3,50].
7. Matrix Factorization (MF): As a latent factor model, the matrix decomposition method mainly decomposes the user-item matrix into user and product latent factor matrices [16]. These two latent factor matrices are then inner-producted to calculate the predicted user-item ratings.
8. Non-Negative Matrix Factorization (NMF): Similar to the MF method, NMF is a matrix decomposition method under the restriction that all elements of the matrix are non-negative [51]. As the true rating of users is non-negative, NMF ensures that the decomposed latent factors are also non-negative.
9. Singular Value Decomposition (SVD++): Singular Value Decomposition (SVD) is a common matrix decomposition technique. In the collaborative filtering method, SVD decomposes the user feature matrix and item feature matrix to predict the ratings, based on the existing user-item matrix. SVD++ [47], an extension of SVD, adds users' latent factors to the user-item matrix to increase prediction accuracy.
10. Aspect-aware Latent Factor Model (ALFM): The latent topics are extracted from reviews by using the aspect-aware topic model (ATM) to model users' preferences and items' features in different aspects [52]. Then, based on the latent factors, the aspect importance of a user towards an item is evaluated and linearly combined with the aspect ratings in the ALFM method to calculate the overall ratings.

#### 4.3.2. Effect of Aspect Number on Rating Prediction

This experiment set and compared the different aspect numbers and parameters of the CNN to conduct the ADLRP method, as proposed in this study. Based on the experimental results, the best aspect number and best CNN parameters were selected. The aspect number was set to 1, 4, 6, 10, 15, 20, 25, 35, 45, etc. The parameters of the CNN include a filter (i.e.,  $F$ ), strides (i.e.,  $S$ ), the activation function of the convolutional layer (i.e.,  $G_{conv}$ ), activation function of fully connected layer (i.e.,  $G_{fc}$ ), and optimizer (i.e.,  $O$ ).

At the beginning of this experiment, we set the parameters as  $(F = 2, S = 1, G_{conv} = \tan H, G_{fc} = ReLu, O = Adam)$  and fixed the convolutional kernel size of the CNN to  $2 * 2$ . The following four steps were used to determine the optimal parameter values of the CNN.

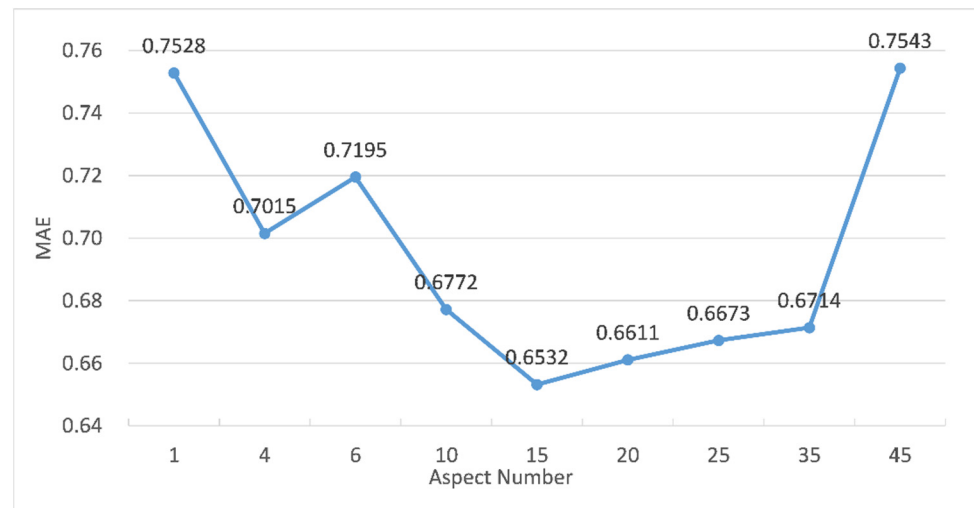
1. Assume that the number of input aspects was  $n$ . The number of filters  $F$  was adjusted from 2 to  $n$  and increased to the power of 2. Based on the prediction accuracy, the optimal number of filters was decided.
2. The moving pace  $S$  was set from 1 to 6 to determine the optimal moving pace.
3. For the activation functions of the convolutional layer and fully connected layer, three activation functions: *Sigmoid*, *tanh*, and *ReLU*, were set, respectively. The best activation function was decided based on the experimental results.
4. The optimizers were set as *Adadelta*, *Adagrad*, *SGD*, and *Adam*, respectively. The optimal optimizer was decided based on the experimental results.

Based on the results of several experiments, the optimal parameters of the CNN in different aspects are shown in Table 1. The ADLRP method has the lowest MAE and RMSE when the aspect number is set as 15. The parameters of the CNN are  $F = 2$ ,  $S = 4$ ,  $G_{conv} = \tan H$ ,  $G_{fc} = ReLu$ ,  $O = Adam$ .

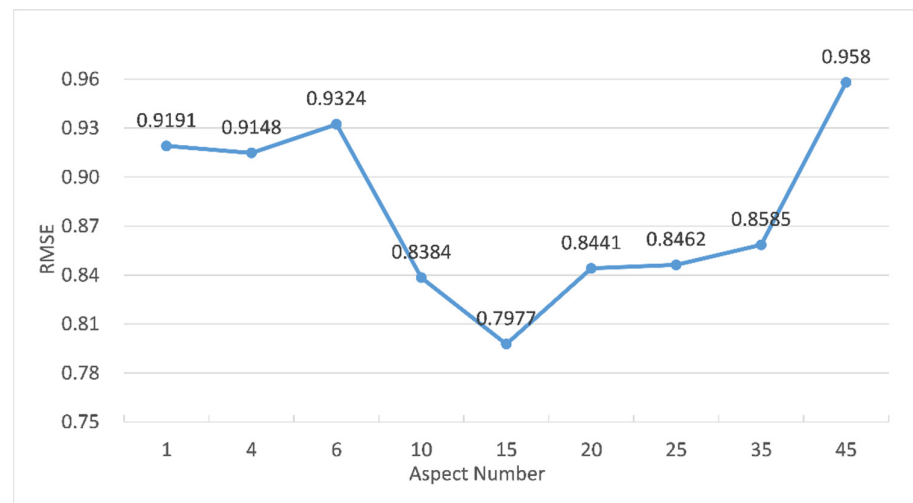
**Table 1.** Optimal combination of parameters for each aspect number.

Aspect Number	<i>F</i>	<i>S</i>	<i>G<sub>conv</sub></i>	<i>G<sub>fc</sub></i>	<i>O</i>	MAE	RMSE
1	2	4	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.7528	0.9191
4	4	2	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.7015	0.9148
6	6	2	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.7195	0.9324
10	2	2	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.6772	0.8384
15	2	4	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.6532	0.7977
20	2	5	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.6611	0.8441
25	1	4	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.6673	0.8462
35	1	5	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.6714	0.8585
45	45	5	<i>tanH</i>	<i>ReLu</i>	<i>Adam</i>	0.7543	0.9580

Figures 4 and 5 show the MAE and RMSE results of predicted ratings in different numbers of aspects, respectively. When the aspect number is 1, the ADLRP method does not consider the aspect features in reviews, and only analyzes the implicit sentiment and semantic features in reviews for predictions. In this setting, the rating prediction result is the worst. When the ADLRP method considers the aspect features, the prediction accuracy can be gradually improved. Therefore, analyzing the aspects in reviews helps to improve prediction accuracy. The ADLRP method has the best prediction results when the number of aspects is set as 15. Experimentation found that when the number of aspects is not considered, or is less, the features of user preferences and product evaluation may not be sufficiently clear, and the prediction accuracy will be low. As the number of aspects increases, the MAE and RMSE also rise gradually, signifying that too many aspects may cause user preferences and product evaluation features to contain less relevant information, which in turn may impede improvement in prediction accuracy.

**Figure 4.** MAE value under different aspect numbers.





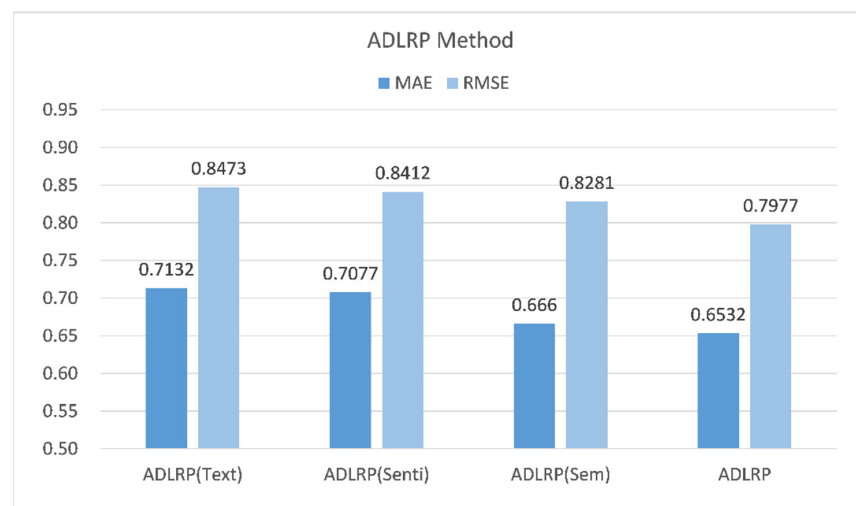
**Figure 5.** RMSE value under different aspect numbers.

#### 4.3.3. Effects of Aspect, Sentiment, and Semantic Features on Rating Prediction

This experiment explored the effects of aspect, sentiment, and semantic features on the rating prediction method (i.e., ADLRP method), and then compared the results of MAE and RMSE. Based on the experimental results in Section 4.3.2, the number of aspects for the ADLRP method was set as 15. The ADLRP method was the basis for comparing the following four features.

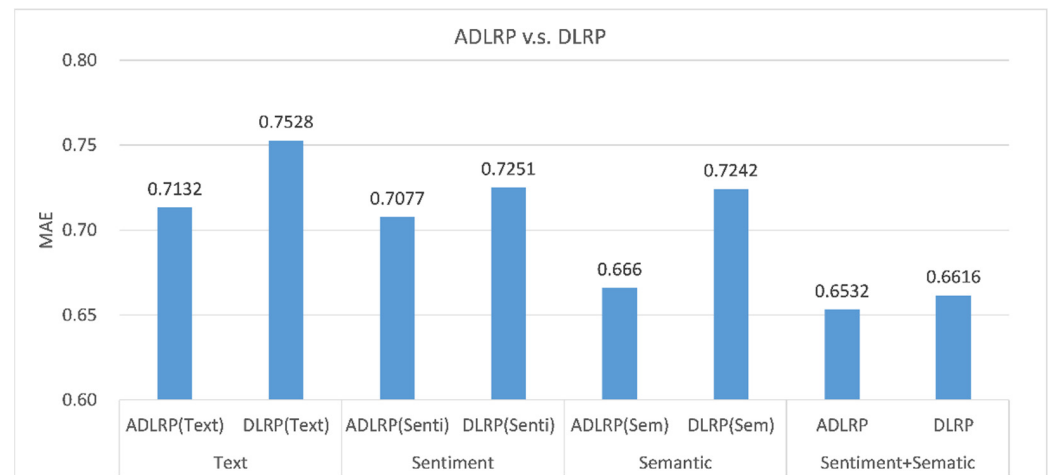
1. ADLRP (Text): This method only considers aspect vectors for rating predictions.
2. ADLRP (Sem): This method integrates the aspect vectors (i.e., Section 3.2) and aspect semantic vectors (i.e., Section 3.5) for rating prediction.
3. ADLRP (Senti): This method integrates the aspect vectors (i.e., Section 3.2) and aspect sentiment vectors (Section 3.3.2) for rating prediction.
4. ADLRP: This method integrates aspect vectors, sentiment vectors, and semantic vectors, and uses a CNN for rating prediction.

According to Figure 6, ADLRP (Text) only considers aspect features, and its rating prediction accuracy is the lowest. The prediction results of ADLRP (Sem) are better than those of ADLRP (Senti), which represents that aspect semantic features have a greater influence on the rating prediction method than aspect sentiment features do. The ADLRP method, which considers aspect, sentiment, and semantic features, helps to improve prediction accuracy.

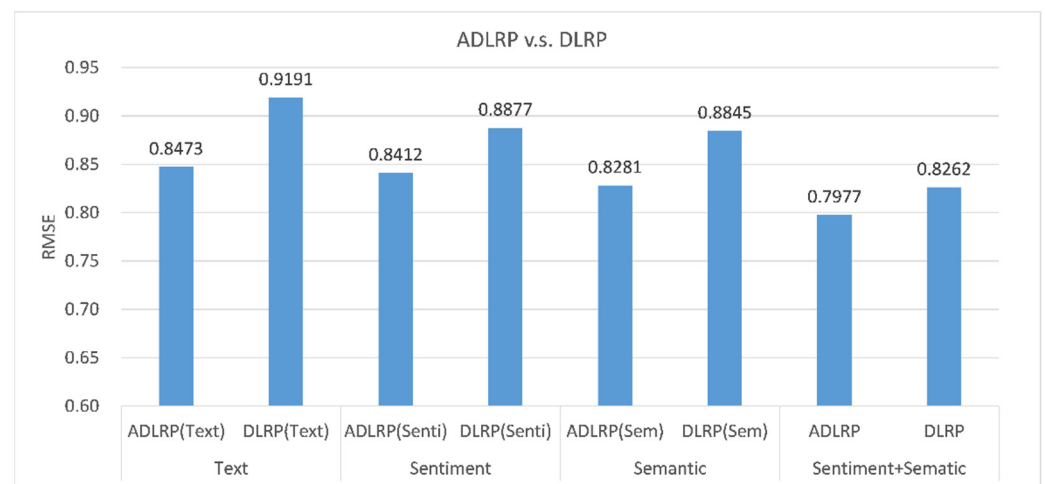


**Figure 6.** Comparison of semantic and sentiment features in consideration of aspect features.

Figures 7 and 8 show the effects of using text, sentiment, semantic, and sentiment in combination with semantic features on the prediction results with and without considering the aspect features, respectively, for comparing the ADLRP-based method with the DLRP-based method. By setting the number of aspects of the ADLRP method as 1, this method is the deep learning rating prediction method (DLRP). The DLRP method does not consider aspect features, but rather employs sentiment and semantic features for rating prediction.



**Figure 7.** Rating prediction MAEs of different features in/without consideration of aspect features.



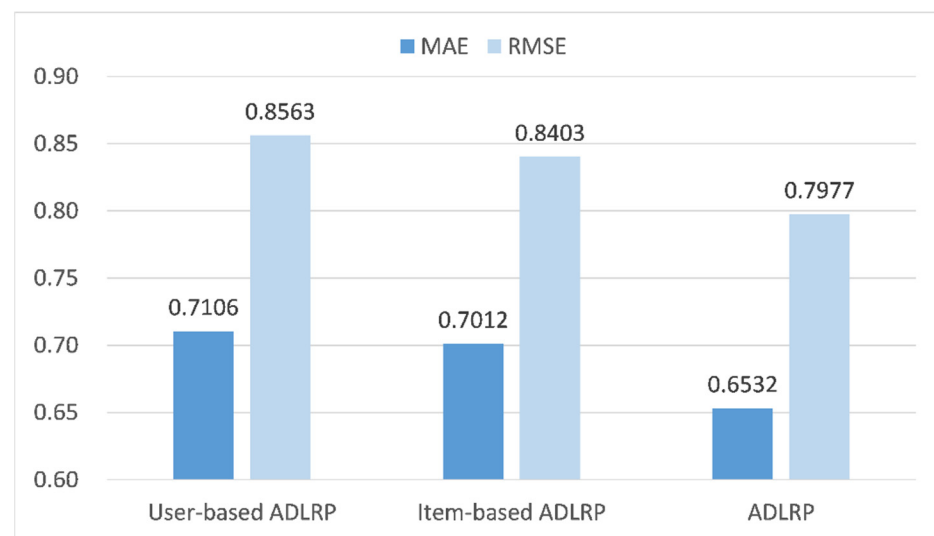
**Figure 8.** Rating prediction RMSEs of different features in/without consideration of aspect features.

Regardless of the features used, it was found that the MAE and RMSE in the ADLRP-based method are better than in the DLRP-based method. Therefore, analyzing the aspect features in reviews helps to improve prediction accuracy. In addition, both the ADLRP and DLRP methods produce the worst prediction results by using only textual features (i.e., ADLRP (Text) and DLRP (Text)), whereas integrating sentiment and semantic features can produce better prediction accuracy. Among the compared methods, the ADLRP method has the best prediction accuracy. Therefore, analyzing the user and product aspects, sentiment, and semantics in the review text can help to better understand user preferences and product evaluation, as well as improve prediction accuracy.

#### 4.3.4. Comparison of User-Based and Product-Based Prediction Methods

The ADLRP method analyzes user reviews versus product reviews. To investigate the influence of user features and item features on the prediction of ratings, this experiment compared the prediction results of the user-based ADLRP method with the item-based

ADLRP method. The user-based ADLRP method analyzes user reviews and extracts user features, including aspect features, sentiment intensity, and semantic features, and inputs the features into the model proposed in this study for rating prediction. Similarly, the item-based ADLRP method extracts features from product reviews for rating prediction. Both methods are based on the ADLRP method; the main difference lies in the features used. The number of aspects was set as 15 in the two methods, and the comparison results are shown in Figure 9.



**Figure 9.** Comparison of user-based and item-based ADLRP methods.

According to Figure 9, we can see that user-based ADLRP has the lowest prediction accuracy, followed by item-based ADLRP; the ADLRP method has the best prediction accuracy. By considering the features extracted from both user and product reviews during predicting rating, we can obtain more accurate user preferences and product ratings, and significantly improve prediction accuracy.

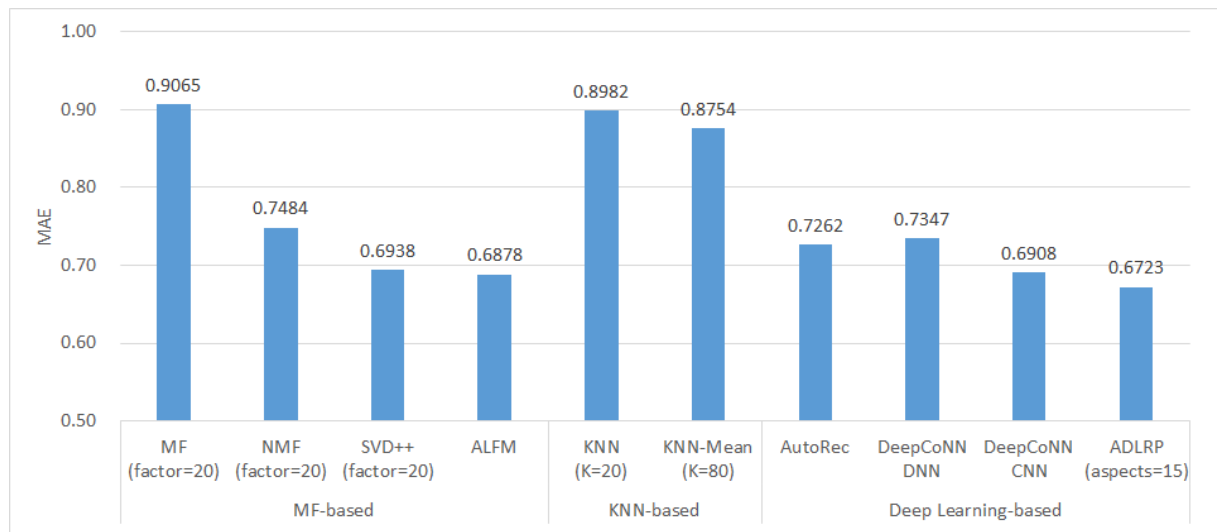
#### 4.3.5. Comparison of Rating Prediction Methods

This experiment compared the ADLRP method, as proposed in this study, with three different types of rating prediction methods, including:

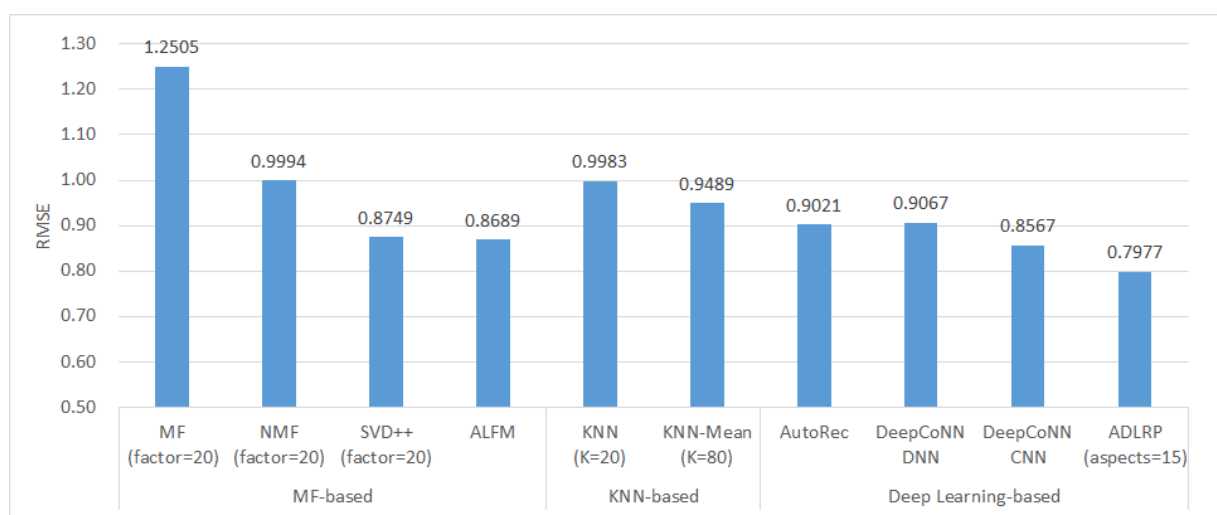
1. Matrix factorization methods: MF, NMF, SVD++, and ALFM.
2. Nearest neighbor methods: KNN and KNN-Mean.
3. Deep learning methods: AutoRec, DeepCoNN-DNN, DeepCoNN-CNN, and ADLRP.

The description of each method is described in Section 4.3.1. In the matrix factorization methods, the number of implied factors of the three MF-related methods was set in the range of 20 to 100, in increments of 10. The results show that the optimal number of implied factors is 20 for all of MF, NMF, and SVD++. In the nearest neighbor methods, the neighbor number  $K$  of the KNN and KNN-Mean methods was set in the range of 20 to 100, in increments of 10 in each experiment. The number of neighbors with the best prediction result was selected as the best parameter. According to the experimental results, the best  $K$  was 20 for KNN and 80 for KNN-Mean. In the deep learning method, the parameters of the AutoRec method were set as follows: the number of neurons: 500, optimizer: *Adam*, and learning rate: 0.001. Both DeepCoNN-DNN and DeepCoNN-CNN use GloVe to convert reviews into word vectors. DeepCoNN-DNN consists of three hidden layers and two fully connected layers, with 200 neurons in each hidden layer, *Adam* as the optimizer, and MSE as the loss function. DeepCoNN-CNN has 2 filters, a convolutional kernel size of 10, and 6 strides, with *tanH* and *ReLu* being the activation functions in the convolutional and fully connected layers, respectively, *Adam* being the optimizer, and MSE being the loss function. The parameters of the ADLRP method, as proposed in this study, are described

in Section 4.3.2. In this experiment, the best parameters of each method were selected to compare the rating prediction results, as shown in Figures 10 and 11.



**Figure 10.** Comparison of MAE in all methods.



**Figure 11.** Comparison of RMSE in all methods.

Among the MF-based methods, ALFM has the best prediction ability. In addition to the original rating matrix, ALFM extracts the latent topics from reviews and predicts ratings based on the combination of aspect importance and aspect ratings. The latent topics and aspect importance help to effectively enhance prediction accuracy. In contrast, the traditional MF-based method only uses the rating matrix for rating prediction, and its prediction accuracy is the worst among all the methods. In the KNN-based methods, the prediction accuracy of the KNN-Mean is slightly better than that of the traditional KNN.

In the deep learning-related methods, the prediction accuracy is AutoRec < DeepCoNN DNN < DeepCoNN CNN < ADLRP. AutoRec only uses the rating data to make predictions and does not analyze the review content; thus, compared with other deep learning-related methods, the prediction accuracy of AutoRec is unsatisfactory. Except for SVD++, AutoRec outperforms the MF-based and KNN-based methods, which only use rating data for predictions. Both DeepCoNN DNN and DeepCoNN CNN predict ratings based only on user and product features, but DeepCoNN CNN outperforms DeepCoNN DNN; this also indicates that a CNN can better improve the accuracy of rating predictions compared

to a deep neural network (DNN). The proposed method, ADLRP, can produce the best prediction results, as it analyzes the implicit aspects, sentiment, and semantics in reviews, which helps to build more accurate user preference features and product evaluation features. CNN can effectively improve the accuracy of rating prediction. In general, the prediction accuracy of a method that analyzes the implicit features of the review text is better than that of a method that only uses the rating data. Moreover, deep learning-related methods also help to improve rating prediction accuracy.

## 5. Conclusions and Future Studies

Due to the extremely large amount of review data on the Internet, users need to spend a lot of time analyzing such data, which reduces the efficiency of purchasing decisions. A recommender system can automatically analyze user preferences and recommend items that may be of interest to future users to solve the problem of information overload. In addition, the textual reviews may imply users' preferences for different topics or aspects, and the sentiment and semantics implied by the text may also express user preferences. In addition to the rating values, more user preferences and implied product features can be extracted from reviews. By accurately analyzing user preferences, the accuracy of rating prediction can be improved. Therefore, this study proposed the aspect-based deep learning rating prediction method (ADLRP), which consists of four main components: aspect detection, sentiment analysis, semantic analysis, and rating prediction. By analyzing user and product reviews, we built feature vectors, including aspect, aspect sentiment, and aspect semantics, which were integrated and trained by a CNN to generate user latent factors and product latent factors. Finally, this study used the MF method to predict the ratings of products that users have not rated. The experimental results show that the best prediction accuracy was achieved by using aspect, aspect sentiment, and aspect semantic features for rating predictions; this also means that integrating multiple features can help build more accurate user preference features and product evaluation features, thereby improving the accuracy of rating prediction, which enables merchants to better understand users' preferences and provide them with more accurate product recommendations.

As the aspect analysis results affect the accuracy of rating prediction, this study will be extended in the future regarding aspect analysis. For example, we will analyze the importance of different aspects to users or products and set the corresponding weights automatically. In addition, semantic features may affect the results of rating prediction. In the future, this study will adopt different methods to analyze the semantic features in reviews, such as Embeddings from Language Models (ELMo) or BERT. In the future, we will analyze the importance of semantic features and sentiment features to users and provide different weights to build an accurate user preference feature vector. As there may be relationships between user behaviors or products, the graphic neural network method will be used to analyze the relations between users and products. By integrating the results of the review analysis and relationship analysis, prediction accuracy will be improved.

**Author Contributions:** Conceptualization, methodology and writing—review and editing, C.-H.L.; methodology, software, and validation, K.-C.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Ministry of Science and Technology of Taiwan, grant number MOST 109-2410-H-033-011-MY2.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <https://jmcauley.ucsd.edu/data/amazon/> (accessed on 31 July 2020).

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

## References

1. Konstan, J.A.; Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R.; Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News. *Commun. ACM* **1997**, *40*, 77–87. [\[CrossRef\]](#)
2. Herlocker, J.L.; Konstan, J.A.; Borchers, A.; Riedl, J. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 15–19 August 1999; pp. 230–237.
3. Sarwar, B.; Karypis, G.; Konstan, J.; Reidl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the Tenth International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
4. Melville, P.; Mooney, R.J.; Nagarajan, R. Content-boosted collaborative filtering for improved recommendations. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, AB, Canada, 28 July–1 August 2002; pp. 187–192.
5. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *Mach. Learn. Res.* **2003**, *3*, 993–1022.
6. Jo, Y.; Oh, A.H. Aspect and sentiment unification model for online review analysis. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 815–824.
7. Poria, S.; Cambria, E.; Gelbukh, A.F. Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network. *Knowl.-Based Syst.* **2016**, *108*, 42–49. [\[CrossRef\]](#)
8. Li, Q.; Li, X.; Lee, B.; Kim, J. A Hybrid CNN-Based Review Helpfulness Filtering Model for Improving E-Commerce Recommendation Service. *Appl. Sci.* **2021**, *11*, 8613. [\[CrossRef\]](#)
9. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In Proceedings of the 3rd International Workshop on Machine Learning Methods for Recommender Systems, Houston, TX, USA, 27–29 April 2017.
10. 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; pp. 425–434.
11. Liu, B. Sentiment Analysis and Opinion Mining. *Synth. Lect. Hum. Lang. Technol.* **2012**, *5*, 1–167. [\[CrossRef\]](#)
12. Tang, D.; Qin, B.; Liu, T. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 1014–1023.
13. Farkhod, A.; Abdusalomov, A.; Makhmudov, F.; Cho, Y.I. LDA-Based Topic Modeling Sentiment Analysis Using Topic/Document/Sentence (TDS) Model. *Appl. Sci.* **2021**, *11*, 11091. [\[CrossRef\]](#)
14. Bradley, M.M.; Lang, P.J. *Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings*; University of Florida: Gainesville, FL, USA, 1999; pp. 25–36.
15. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical Attention Networks for Document Classification. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
16. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [\[CrossRef\]](#)
17. Khan, Z.Y.; Niu, Z.; Sandiwarno, S.; Prince, R. Deep learning techniques for rating prediction: A survey of the state-of-the-art. *Artif. Intell. Rev.* **2021**, *54*, 95–135. [\[CrossRef\]](#)
18. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 25–29 October 2014; pp. 1746–1751.
19. Wang, J.; Yu, L.-C.; Lai, K.R.; Zhang, X. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 225–230.
20. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
21. Bengio, Y.; Simard, P.; Frasconi, P. Learning Long-term Dependencies with Gradient Descent is Difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [\[CrossRef\]](#)
22. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
23. Schuster, M.; Paliwal, K.K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [\[CrossRef\]](#)
24. Tang, D.; Qin, B.; Liu, T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1422–1432.
25. Wang, Y.; Huang, M.; Zhao, L. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
26. Cao, R.; Zhang, X.; Wang, H. A Review Semantics Based Model for Rating Prediction. *IEEE Access* **2020**, *8*, 4714–4723. [\[CrossRef\]](#)
27. Liu, W.; Lin, Z.; Zhu, H.; Wang, J.; Sangaiah, A.K. Attention-Based Adaptive Memory Network for Recommendation with Review and Rating. *IEEE Access* **2020**, *8*, 113953–113966. [\[CrossRef\]](#)
28. Hinton, G.E.; McClelland, J.L.; Rumelhart, D.E. Distributed representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*; David, E.R., James, L.M., Group, C.P.R., Eds.; MIT Press: Cambridge, MA, USA, 1986; Volume 1, pp. 77–109.
29. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Seattle, WA, USA, 18–21 October 2013; pp. 1059–1069.



30. Porcel, C.; Moreno, J.M.; Herrera-Viedma, E. A Multi-Disciplinary Recommender System to Advice Research Resources in University Digital Libraries. *Expert Syst. Appl.* **2009**, *36*, 12520–12528. [\[CrossRef\]](#)
31. Kamyab, M.; Liu, G.; Adjeisah, M. Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis. *Appl. Sci.* **2021**, *11*, 11255. [\[CrossRef\]](#)
32. Dang, C.N.; Moreno-García, M.N.; De la Prieta, F. Using Hybrid Deep Learning Models of Sentiment Analysis and Item Genres in Recommender Systems for Streaming Services. *Electronics* **2021**, *10*, 2459. [\[CrossRef\]](#)
33. Baccianella, S.; Esuli, A.; Sebastiani, F. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 17–23 May 2010; pp. 2200–2204.
34. Cambria, E.; Olshe, D.; Rajagopal, D. SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1515–1521.
35. Dong, Z.; Dong, Q.; Hao, C. HowNet and the Computation of Meaning. In Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, Beijing, China, 23–27 August 2010; pp. 53–56.
36. Hamouda, A.; Rohaim, M. Reviews classification using SentiWordNet lexicon. *Online J. Comput. Sci. Inf. Technol. (OJCSIT)* **2011**, *2*, 120–123.
37. Baltrunas, L.; Ludwig, B.; Ricci, F. Matrix factorization techniques for context aware recommendation. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 301–304.
38. Zhang, X.; Liu, H.; Chen, X.; Zhong, J.; Wang, D. A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness. *Inf. Sci.* **2020**, *519*, 306–316. [\[CrossRef\]](#)
39. Wu, D.; Shang, M.; Luo, X.; Wang, Z. An  $L_1$ -and- $L_2$ -Norm-Oriented Latent Factor Model for Recommender Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–14. [\[CrossRef\]](#)
40. Luo, X.; Yuan, Y.; Chen, S.; Zeng, N.; Wang, Z. Position-Transitional Particle Swarm Optimization-incorporated Latent Factor Analysis. *IEEE Trans. Knowl. Data Eng.* **2020**, *1*. [\[CrossRef\]](#)
41. Wu, D.; He, Q.; Luo, X.; Shang, M.; He, Y.; Wang, G. A Posterior-neighborhood-regularized Latent Factor Model for Highly Accurate Web Service QoS Prediction. *IEEE Trans. Serv. Comput.* **2019**, *1*. [\[CrossRef\]](#)
42. Wu, D.; Luo, X.; Shang, M.; He, Y.; Wang, G.; Wu, X. A Data-Characteristic-Aware Latent Factor Model for Web Services QoS Prediction. *IEEE Trans. Knowl. Data Eng.* **2020**, *1*. [\[CrossRef\]](#)
43. Zhang, Y.; Ai, Q.; Chen, X.; Croft, W.B. Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources. In Proceedings of the ACM Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1449–1458.
44. Yin, W.; Schütze, H.; Xiang, B.; Zhou, B. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 259–272. [\[CrossRef\]](#)
45. Li, F.; Wu, B.; Xu, L.; Shi, C.; Shi, J. A fast distributed stochastic Gradient Descent algorithm for matrix factorization. In Proceedings of the 3rd International Conference on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, New York, NY, USA, 24 August 2014; pp. 77–87.
46. Zhuang, Y.; Chin, W.-S.; Juan, Y.-C.; Lin, C.-J. A fast parallel SGD for matrix factorization in shared memory systems. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 249–256.
47. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
48. Sedhain, S.; Menon, A.K.; Sanner, S.; Xie, L. Autorec: Autoencoders Meet Collaborative Filtering. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 111–112.
49. 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. [\[CrossRef\]](#)
50. Park, Y.; Park, S.; Jung, W.; Lee, S.-G. Reversed CF: A Fast Collaborative Filtering Algorithm Using a K-nearest Neighbor Graph. *Expert Syst. Appl.* **2015**, *42*, 4022–4028. [\[CrossRef\]](#)
51. Zhang, S.; Wang, W.; Ford, J.; Makedon, F. Learning from incomplete ratings using non-negative matrix factorization. In Proceedings of the SIAM International Conference on Data Mining, Bethesda, MD, USA, 20–22 April 2006; pp. 549–553.
52. 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; pp. 639–648.