Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9015

MENN: A Hybrid Model for User Preference Mining

Yaoxuan Guo[1]. Yuchen Xu[2] and Jiaxiang Ge[1]
[1]School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
[2]College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China

How to cite this paper: Guo, Y. X., Xu, Y. C., & Ge, J. X. (2025). MENN: A Hybrid Model for User Preference Mining. Journal of Computer Science and Frontier Technologies, 1(2), 52-66. ISSN Print: 3104-4204; ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9015
Published: 2025-10-17

Abstract

User preference mining uses rating data, item content or comments to learn additional knowledge to support the prediction task. For the use of rating data, the usual approach is to take rating matrix as data source, and collaborative filtering as the algorithm to predict user preferences. Item content and comments are usually used in sentiment analysis or as auxiliary information for other algorithms. However, factors such as data sparsity, category diversity, and numerical processing requirements for aspect sentiment analysis affect model performance. This paper proposes a hybrid method, which uses the deep neural network as the basic structure, considers the complementarity of text and numeric data, and integrates the numeric and text embedding into the model. In the construction of text-based embedding, extracts the text summary of each text-based review, and uses the Doc2vec to convert the text summary into multi-dimensional vector. Experiments on two Amazon product datasets show that the proposed model consistently outperforms other baseline models, achieving an average reduction of 15.72% in RMSE, 24.13% in MAE, and 28.91% in MSE. These results confirm the effectiveness of our proposed method for learning user preferences.

Keywords

User preference; Numeric; Text; Neural networks

1. Introduction

The exponential growth of online platforms has created an unprecedented challenge in personalized recommendation systems. With billions of users generating massive amounts of heterogeneous data daily, accurately mining user preferences has become critical for enhancing user experience and platform engagement[1].

Traditional user preference mining methods primarily rely on single-type data sources , such as rating data or textual reviews, which often fail to fully utilize the rich information embedded in multi-source heterogeneous data. Existing research demonstrates that user preference expressions typically manifests in multiple forms of data: numerical data (such as ratings, click counts, browsing duration, etc.) can quantify user behavioral intensity, while textual data (such as reviews, tags, descrip-

tions, etc.) can reveal semantic features and emotional tendencies of user preferences. However, how to effectively fuse these heterogeneous data to obtain more accurate user preference representations still faces numerous challenges[2].

Current multimodal fusion methods generally suffer from noise interference when processing textual data. User reviews often contain substantial redundant information irrelevant to core preferences. This noisy text not only fails to contribute effective preference information but may also mislead the model's learning process, reducing the accuracy of preference mining. Furthermore, existing methods lack deep theoretical analysis and effective technical approaches in fusion strategies for numerical and textual embeddings, making it difficult to fully leverage the complementary advantages of different modal data[3].

To address these issues, this paper proposes the Multi-form information Embedding Neural Network (MENN), a deep learning model for mining user preferences. The main contributions of this method include: First, a hybrid method is established to learn user preference knowledge from numerical embedding and textual embedding. Second, in the process of constructing a textual embedded vector, we develop a novel way to weed out the redundant texts which do not contribute to the main intent of the user reviews and may confound the model results. Third, noting the complementarity of textual and numerical data, the textual embedding and numerical embedding are fused into a joint embedding representation vector to facilitate analysis of the model.

2. Related Works

This study proposes a deep neural network method that effectively integrates het erogeneous information, specifically numerical embeddings and textual embeddings, for user preference mining. Focusing on the core aspects of this research, we explain related works from three perspectives: predicting user preferences using rating data, predicting user preferences using text data, and user preference mining based on deep learning[4].

2.1. Using Rating Data to Predict User Preferences

Collaborative Filtering (CF) is one of the most widely used methods in recommen dation systems, predicting user preferences by analyzing the user-item rating matrix. Its core assumption is that if two users have similar rating patterns for certain items, their preferences for other items are likely to be similar. CF methods generate recommendations by finding similar users or items, with user similarity typically measured using Pearson correlation coefficient and cosine similarity. In recent years, improving the accuracy of rating predictions has become a research focus, and modeling rating preference behavior significantly impacts recommendation performance. Rating data contains user preferences and forms the basis for user preference analysis. However, the number of rating datasets is limited, generating rating data incurs costs, and data sharing may violate user privacy. Therefore, many stud-

ies attempt to create comprehensive datasets through data genera tion and use synthetic datasets to evaluate models. Since synthetic datasets are based on original datasets, they inevitably bear the characteristics of the original datasets. Moreover, it is unclear whether unknown datasets used for model training and testing will exhibit different characteristics in actual situations and lead to different model outcomes. Thus, a relatively reasonable measure is to increase the categories and dimensions of the data, such as adding comment text corresponding to the numerical data[5].

2.2. Using Text Data to Predict User Preferences

The core of using natural language processing techniques to predict user preferences lies in employing sentiment analysis methods to deeply explore the emotional tendencies and personal preferences contained in user reviews. The application value of sentiment analysis in recommendation systems has been widely recognized, as user-generated text reviews constitute a vast repository of emotional opinions about products and services. Deep learning models such as Convolutional Neural Networks (CNN), Recurrent Neu ral Networks (RNN), and Bidirectional Long Short-Term Memory Networks (Bi-LSTM) have demonstrated excellent performance in sentiment prediction tasks for product re views. Furthermore, the advent of pre-trained language models like BERT has significantly. enhanced the accuracy of text sentiment analysis. In specific scenarios of e-commerce recommendations, models based on Transformers can effectively parse the complex semantics in user reviews, providing more precise sentiment judgments for product rating predictions. Notably, compared to rating data, text reviews carry richer semantic information, offering more detailed, comprehensive, and reliable insights into user preferences for recommendation systems, thereby aiding in constructing more refined representations of user preferences[6].

2.3. User Preference Mining Based on Deep Learning

Deep learning technology has catalyzed transformative advancements in the field of user preference mining by leveraging hierarchical feature learning capabilities to au tomatically discover complex user-item interaction patterns. Through multi-layer neural network architectures with increasing levels of abstraction, deep learning models can learn sophisticated representations that capture both explicit and implicit preference signals, effectively modeling the inherent nonlinearities and high order interactions present in user behavior data. These deep hidden neural architectures demonstrate exceptional capabilities across diverse recommendation scenarios, including automated extraction of latent user profiles and item embeddings, comprehensive understanding of complex nonlinear user item dependencies, efficient scalability for large-scale product recommendation systems, and accurate prediction of sequential user actions in session based recommendation applications. Currently, mainstream deep learning recommendation models employ various ar-

chitectures, such as multilayer perceptrons, convolutional neural networks, recurrent neural networks, and the more advanced Transformer models. In recent years, graph neural networks have become a key technological breakthrough, regarded as a core tool for solving real-world problems such as recommendation systems, drug development, and traffic prediction. Additionally, user preference mining methods that incorporate refined sentiment analysis have made significant progress. These methods leverage the powerful feature learning capabilities of deep neural networks to excel in accurately extracting user features and handling complex emotional expressions, effectively enhancing model performance.

Existing research either focuses on deep exploration of a single modality or uses simple concatenation or weighted summation for fusion. This study proposes a hybrid approach that achieves deeper cross-modal information interaction within a unified frame work by establishing a collaborative learning mechanism for numerical and text embed dings, surpassing the limitations of traditional methods that process rating data and text data independently.

3. Approach

On online user preference mining, in addition to numerical rating information, text based reviews are another good means to mine user preferences. The user's textual comments on the item complement the numerical rating, reflecting the user's preference for the product. Therefore, this paper considers the joint effects of numerical embedding and textual embedding on deep neural networks. At the same time, by taking the summary of the user comments as the data embedding bridge, it effectively avoids the possible interference of some ambiguous and irrelevant sentences in the original comment corpus on the final results of the model. In what follows, we examine the model's overall structure, the numerical information embedding, and the textual information embedding and the individual user preference.

3.1. Model Structurs

Figure 1 shows the overall architecture of the proposed model. Using a deep neural network architecture, both numerical embedding and textual embedding are considered in the embedding layer of the model. The data used for numerical embedding includes user ID and item ID. The textual embedding uses textual comment data, which correspond to the user IDs and item IDs. The label value of the model adopts the corresponding user's rating value on the item. The size of the label reflects the user's preference for the product. The larger the label value, the more the user prefers the product. Numerical embedding learns the representation of user preference items through user and item information, while textual embedding learns user preference through textual comment information. Since numerical embedding and textual embedding are complementary, text embedding explains the reasons why users rate the items in greater depth. Combining numerical embedding

and text embedding improves the performance of the model in predicting user preferences.

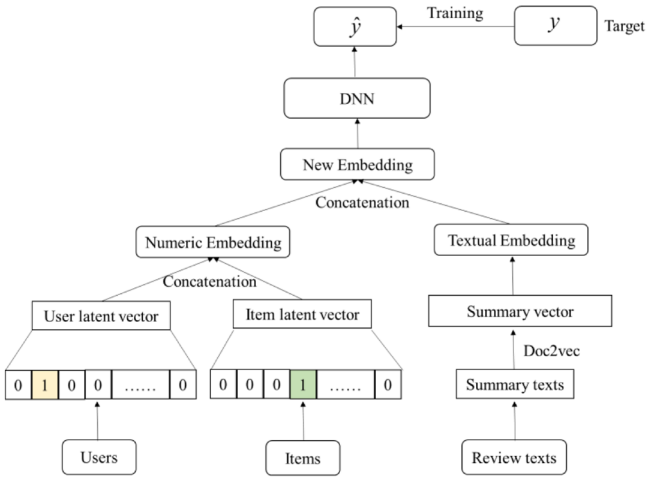

Figure 1. Structure of research approach.

Further, the proposed method does not simply divide the comment text into words and then map the words onto the data lexicon and embed them into the model. In the actual text comment section, each comment reflects the user’s preference for a certain product. Since we focus on extracting user preferences from the comments, compared to the complicated user comment text, the summary of the comment text clarifies the main idea of the comment, and removes other sentences that are not related to the main idea of the text, so it is conducive for reflecting user preferences. Therefore, the first step of textual embedding for us is to obtain a summary of the comment based on the user’s textual comments. Then, the Doc2vec algorithm is used to convert the summary into a multi-dimensional vector, which represents the user’s text comment information about the item. After obtaining the multi-dimensional vector of the textual comment, it is combined with the numerical embedding vector to embed the deep neural network for model training. This approach considers the relationship between user comments and scoring values, as well as the context between words in the comment text, and uses basic knowledge about users and items in the training of the model.

In short, the proposed method comprises three steps. The first step constructs the numerical embedding of the model. The second step builds the textual embedding of the model. The third step combines the numerical embedding with textual embedding to form a new embedding layer. This embedding layer data is fed to the DNN model for training to predict the user’s preference.

3.2. Numeric Information Embedding

Figure 2 shows how the user and item information are converted into the embedded vectors. This section describes how to convert numerical information into embedding vectors, a process consisting of two stages: encoding the numeric data and then

converting it into embedding vectors.

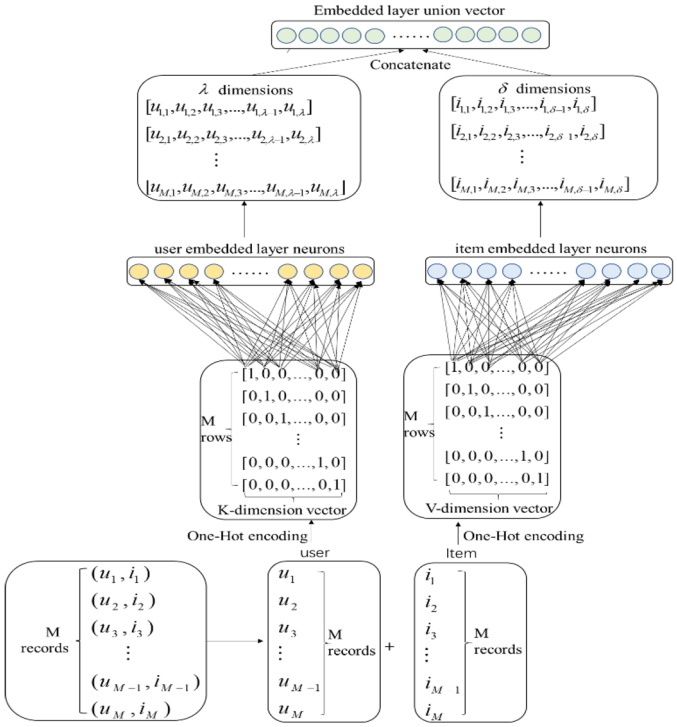

Figure 2. Numerical information embedding process.

3.2.1 Numeric Information Embedding

Encoding numeric data is an important part of numeric information embedding. In machine learning applications, features are sometimes not always continuous, and for such features, it is often necessary to digitize them. For example, a field with an attribute of "region" may contain "Asia, Europe, US". To digitize such information, the most direct way is to serialize each feature as a value, namely, converting the three features contained in the "region" field into [0,1,2]. However, such feature processing cannot be fed directly into the machine learning algorithm. Hence, if such a numeric feature processing method is used in the model calculation, it will affect the model's performance. In this case, the value "0,1,2" represents three features. If they are applied directly to the computation, the value of the numeric features will affect the result.

For example, the commonly used distance or similarity calculation is based on Euclidean distance, rather than directly using a single value.One-Hot encoding is a way to extend the value of discrete features to Euclidean space. The first step of information coding is to number the users and items; each user and item generating a unique ID number. Second, One-Hot encoding is used to convert the user and item information into multi-dimensional vectors whose dimensions relate to the number of users and items. One-Hot encoding, or one-bit effective encoding, uses N-bit status registers to encode the N states. Each state has a registered bit, and only one bit

is valid at a time. The One-Hot encoding of the user and the item requires that they be mapped to integer values separately. Each integer is represented as a binary vector, except that the value at the position of the integer index is marked as 1, and the value of the other position is 0.

3.2.2. Convert Numeric Information to Embedding Vectors

The One-Hot encoding of the users and items is converted into multi-dimensional vectors to facilitate the model computation. With more users and items, the vector dimen sion after One-Hot encoding increases. For example, when the number of users is 105, its dimensions of the encoded vector exceed 105, with most vector elements being 0. Clearly, the encoded data is too sparse, which causes it to overuse resources. As such, an embedding layer needs to be introduced to map the high-dimensional sparse coding vectors to the low-dimensional dense feature representation vector. We use a fully connected layer for the embedding layer, and set the dimension of the embedding vector, that is, how many "factors" do each user and item need to assign, which will affect the vector representation of the user and item. The encoded vectors of the user and the item are transformed into embedded vectors through a full connection layer, and each embedded vector is updated during model training.

In the numerical data embedding, we used numerical data including user ID, item ID, and user rating. The rating serves as the label for the deep neural network model. Each record in the original dataset contains the user ID and the item ID, with the user for each record denoted as ui. The item information for each record is represented as ij. Suppose the dataset contains M records. In the original dataset, a user may correspond to multiple items, and an item may also correspond to multiple users, meaning the number of users and items is not equal to the size of the dataset M. If the number of users is K and the number of items is V, then the dimensions of the users and items after numerical coding are K and V, respectively. Two parallel embedding layers are used for the user and item embeddings. The user and item embedding vectors are then combined at the top layer to form a new joint embedding vector.

$$\begin{aligned} y_i^{(1)} &= \sigma(W^{(1)}u_i + b^{(1)}) \\ y_i^{(2)} &= \sigma(W^{(2)}u_i + b^{(2)}) \\ y_i^{(\lambda)} &= \sigma(W^{(\lambda)}u_i + b^{(\lambda)}) \end{aligned} \qquad 01$$

where $\mathbf{y_i^{(1)}}$, $\mathbf{y_i^{(2)}}$represent the output results of different neurons in the embedded layer after the input ui is processed by the embedded layer, and λ is the number of neurons in the user's em bedded layer. σ denotes the activation function, W(1), W(2) and W(λ) are the corresponding weights of the different neurons, while b(1), b(2) and b(λ) are the bias quantities.

3.3. Textual Information Embedding

3.3.1. Summarization of Comment Text

First, we generate a summary of the user comments. The underlying assumption is that the summary represents the user’s feelings about an item, which potentially reflects the user’s preference of the item. The summary is designed to transform the comments into short texts containing key information, and its generation method can be divided into an extraction method and an abstraction method. The abstractive summary rephrases the key topics of a document using terms that do not appear in the source document. It allows new words and phrases to be generated to form a summary. The abstractive summary method allows new words or phrases to be included, and the summary expression has a higher flexibility. However, due to the complexity and diversity of natural language expressions, some identical words can be combined into several sentences with different meanings. Though the words that form a sentence and the order of the words are similar, sometimes the difference in punctuation or the position of the punctuation may invoke a completely different expressive meaning in the sentence. In the analysis of user preferences, if the abstractive summary generates new words, but the sentence composed of these words does not fully express the original text, or due to differences in grammar, syntax, or punctuation cause a part of the sentence to deviate from the original meaning, this situation will lead to biased results. On the other hand, the extractive summarization method relies on the original text. Extractive summarization depends on mining the features from source sentences and paragraphs and putting them into the classification model to determine whether the input sentence should be placed in the summary. Although the flexibility of the summarization is low, this method has a low error rate in grammar and syntax, which ensures that the generated summary can express the intent of the original text. This way of text summarization has a very good auxiliary role in mining user preferences.

3.3.2 Convert Summarization to Embedding Vectors

To ensure that the user’s preferences expressed in the comment are correctly identi f ied and modeled, we apply the paragraph vector embedding technique of the Doc2vec algorithm after obtaining the summary of the comment, to transform the generated summary text into a vector representation. The algorithm uses a neural network to predict the subsequent words in the sentence through the training of the input words and sen tences, so as to obtain the sentence representation vector. Therefore, after all the comment summaries are generated, they will be treated as new separate documents, and the Doc2vec algorithm is used to convert the independent documents into vector representations. The transformation into vector representations is convenient for different operations on them, and it aids further analysis of the comments represented by these real value vectors.

3.4. Individual User Preference Extraction

After obtaining the textual embedding and numerical embedding data, the two types of embedding vectors are merged to form a new embedding vector, indicating that textual comments and numerical data are complementary in the analysis of a user's preferences, and the two collectively affect the user's preferences. Given the convenience of the on line platform in disseminating and exchanging information, a user's preference may be influenced by users who have previously commented on the network in addition to their owninternal factors. Moreover, the user's preference may be related to temporal factors. The farther away from the user's current comment time, the smaller the impact on the user's preference. Therefore, this paper selects the deep neural networks (DNN) model as the basic architecture and sends the new joint embedding vector to the DNN model for training. The final output represents the user's preference for a certain item.

4. Experimentation

We employ the Tensorflow and Keras frameworks to conduct a series of experiments. Thefollowing contents are carried out using the dataset adopted in the experiment, baseline comparison methods, relevant parameter setting, evaluation metrics, and experimental effect comparison.

4.1. Datasets

We select the data of the Amazon platform, which provides not only a large number of numerical data, but also the textual review of the users on the corresponding products. During the data sampling phase, we employed a stratified random sampling strategy based on the distribution of ratings. As the original dataset is excessively large, for the sake of experimental efficiency, we extracted 50,000 samples from each of the "Electronic" and "Book" Amazon datasets, totaling 100,000 samples. This method ensures that our smaller subset aligns with the original dataset's user rating tendencies. In the data preprocessing phase, we focused on standardization, as well as cleaning punctuation and stop words. Standardization involved converting all text to lowercase and removing HTML tags. The cleaning process eliminated all punctuation and used the standard English stop word list provided by the NLTK library to filter out high-frequency but low-information words. Table 1 describes the variables used in these two datasets, and Table 2 presents the detailed statistical information of the data.

Table 1. Description of the variables in the datasets.

Variable	Type	Description
Reviewer ID	String	ID of the reviewer
Asin	String	ID of the product
Reviewer Name	String	Name of the reviewer
Review text	String	Text of the review
Overall	Float	Rating of the product

Table 2. Detailed statistical information of the datasets.

Statistic	'Electronics' Dataset	'Book' Dataset
Total Reviews	50,000	50,000
Unique Users	43,152	47,881
Unique Items	934	1,012

The fields contained in these two datasets are "reviewer ID" (ID of the reviewer), "asin" (ID of the product), "reviewer Name" (name of the reviewer), "helpful" (helpfulnessrating of the review), "review Text" (text of the review), and "overall" (rating of the product). We supply the data of "reviewer ID", "asin", "overall", and "review Text" to the proposed model. Among them, "reviewer ID", "asin", and "overall" are numerical data, representing the users' ratings on the corresponding products, while "review Text" is textual data, representing the users' specific evaluations of the products. In each round of experimentation, we randomly select 80% of the data as the training set, 10% as the validation set, and the other 10% as the test set. After using the training data for model training, the trained model is used in the test set to evaluate model performance.

4.2. Baseline Methods

To validate the effectiveness of the proposed method in this paper, we selected a series of representative baseline methods for experimental comparison. This set of baselines covers multiple dimensions: it includes classic models like Factorization Machines (FM) and Deep Neural Networks(DNN)for feature interaction and nonlinear learning, as well as advanced models like Deep Factorization Machine (DeepFM) model, which specifically addresses the temporal dynamics of user preferences, and the Stacked Auto encoder as a representative of unsupervised representation learning. These methods were chosen to cover a range from classic recommendation algorithms to cutting-edge deep learning models, aiming to comprehensively evaluate our model's performance.

FM: The factorization machine model constructs the feature combinations by finding the inner product of the hidden variables of each dimension feature. The core idea of FM is to learn a low-dimensional latent vector for each feature and use the inner product of these vectors to efficiently model second-order interactions between all feature pairs. This effectively addresses the feature combination explosion problem encountered by traditional polynomial models when handling large-scale sparse data. In this study, we chose FM as a baseline model for the following reasons: our proposed model aims to capture higher order nonlinear feature interactions through deep networks, while FM focuses on explicit second-order interactions. Therefore, directly comparing with FM can clearly demonstrate the effective gains of our model in learning complex higher-order relationships.

DNN:DNN for ms the foundation of deep learning technology, enabling the automat-

ic learning and extraction of complex nonlinear relationships from data through a network structure with multiple hidden layers. In this study, we use DNN as abase line model for the following reasons: since our proposed model is a deep learning architecture itself, a basic DNN model serves as the most direct and fair comparison.By comparing performance with the DNN, we can clearly assess the real performance improvements brought by specific modules we designed, such as text processing and feature fusion mechanisms, thereby strongly demonstrating the innovative value of our model architecture.

DeepFM: DeepFM is an advanced recommendation model that combines the advantages of Factorization Machines (FM) and Deep Neural Networks (DNN) through a clever end-to-end parallel architecture. In this model, the FM component efficiently learns low order (especially second-order) explicit feature interactions, while the DNN component delves into complex and implicit nonlinear relationships hidden in the data. Both components share the same feature embedding layer, greatly enhancing the model's learning efficiency and expressive power. In this study, DeepFM is chosen as the baseline model primarily due to its successful integration of different levels of feature interactions, which directly aligns with our research goal of integrating heterogeneous features (text and numerical data). By comparing performance with DeepFM, we can clearly evaluate whether our proposed integration strategy offers superiority over this established interaction fusion paradigm.

4.3. Evaluation Metrics

The MeanAbsolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE) are commonly used metrics to evaluate model performance. MAE measures the average value of the absolute error, which reflects the error between the predicted and the actual values. MAE is as follows.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\lvert y_i - \hat{y}_i\rvert \tag{02}$$

MSE sums the square of the difference between the predicted value and the actual value and then averages the sum of the squares. RMSE is the square root of MSE. These two indices are often used as the standard for measuring the prediction results of the model. The MSE and RMSE are written as:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{03}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{04}$$

y_i and $\hat{y}_i$ represent the actual value and the predicted value, respectively, m rep-

resents the number of real value or predicted value involved in the calculation, that is, the data size of the test set.

4.4. Performance Comparison

Our experiments were conducted on Amazon's "Electronic" and "Book" datasets. During the textual embedding process, the abstract of the comment text is extracted and transformed into an embedding vector. The abstract, being a concise summary of the user's comment, captures the core idea of the text. Consequently, embedding the abstract into the model not only enhances its performance but also helps to mitigate the impact of redundant and complex text descriptions on the model. The proposed method introduces a novel Deep Neural Network (DNN) based on multi-form information embedding, which we will refer to as the MENN model for simplicity.

Table 3 shows the evaluation metrics (MAE, MSE, and RMSE) for the models on the "Electronic" and "Book" datasets. The MENN model outperforms all other models, achieving the lowest values across all metrics, with MAE values of 0.16 for both datasets, MSE values of 0.13 (Electronic) and 0.14 (Book), and RMSE values of 0.36 (Electronic) and 0.38 (Book). In comparison, the FM model performs the worst, with the highest MAE, MSE, and RMSE values, indicating its poor prediction accuracy. The DNN and DeepFM models perform better than FM, but still fall short of MENN, with DNN achieving relatively low values of 0.19 (Electronic) and 0.25 (Book) for MAE, and DeepFM showing moderate performance with MAE values of 3.76 (Electronic) and 3.94 (Book). The results highlight that the MENN model, leveraging advanced deep learning techniques, delivers the best predictive performance, while FM and TECF lag behind due to their lack of deep learning integration.

Table 3. Evaluation metrics for experimental models.

Metrics	Dataset	FM	DNN	DeepFM	MENN
MAE	Electronic	4.15	0.19	3.76	0.16
	Book	4.47	0.25	3.94	0.16
MSE	Electronic	18.55	0.18	15.50	0.13
	Book	21.01	0.25	16.46	0.14
RMSE	Electronic	4.30	0.42	3.93	0.36
	Book	4.58	0.50	4.05	0.38

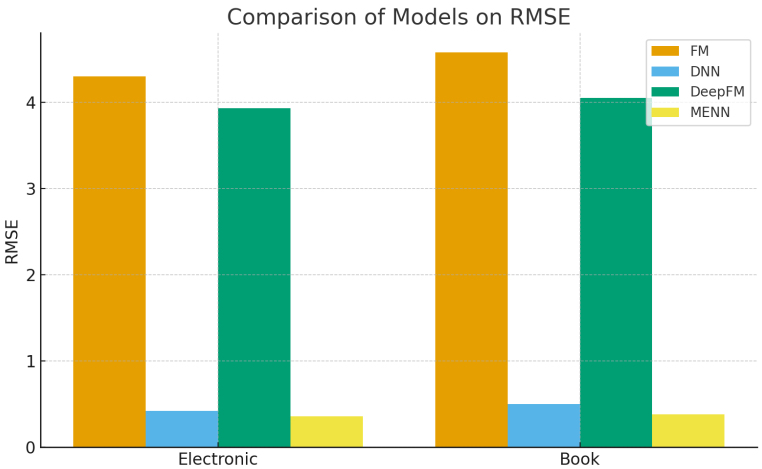

Figure 3. Comparison of Models on RMSE.

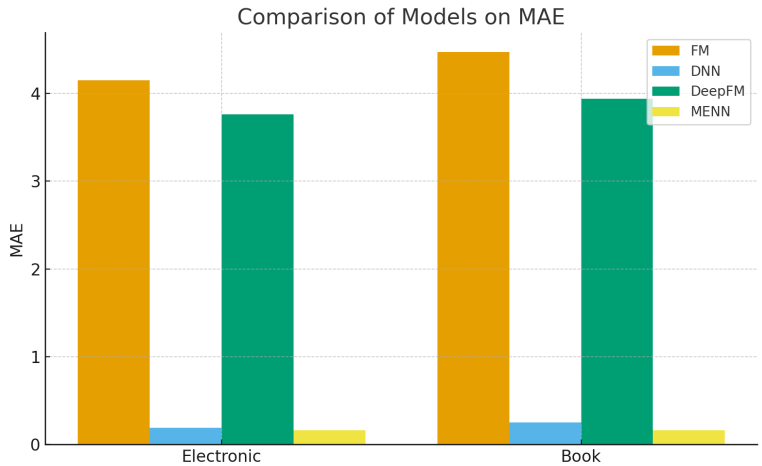

Figure 4. Comparison of Models on MAE.

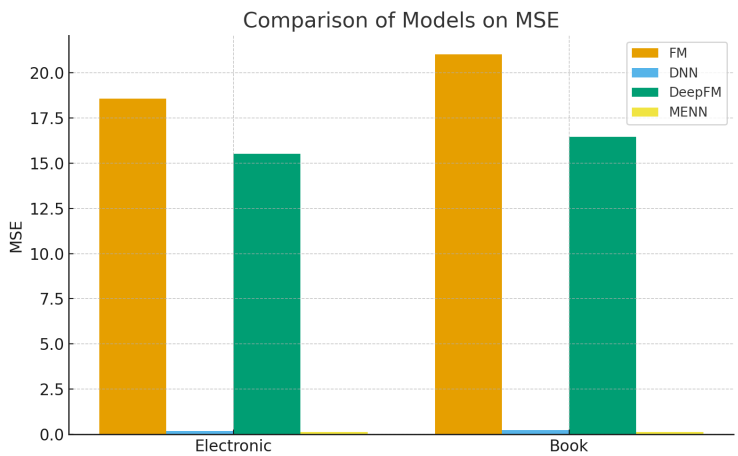

Figure 5. Comparison of Models on MSE.

The bar charts compare the performance of four models—FM, DNN, DeepFM, and

MENN—on the "Electronic" and "Book" datasets using MAE, MSE, and RMSE as evaluation metrics. Across all three metrics, the MENN model consistently achieves the lowest values, indicating superior predictive accuracy and stability. In contrast, the FM model records the highest errors in both datasets, demonstrating poor performance. The DeepFM model performs better than FM due to its partial integration of deep learning but still falls short of MENN. The DNN model shows moderate results, outperforming FM but not reaching MENN's level. Overall, the results clearly show that the MENN model outperforms all baselines, effectively reducing prediction errors and exhibiting the strongest generalization capability.

5. Conclusion

This study proposed the Multi-form information Embedding Neural Network (MENN) model to address the limitations of traditional recommendation systems in mining user preferences from heterogeneous data sources. By integrating both numerical and textual embeddings within a unified deep neural network framework, the MENN model effectively captures the complementary relationships between quantitative behavioral indicators and qualitative semantic information. The introduction of text summarization in the textual embedding process reduces redundant and noisy information, enabling the model to focus on the core user intent expressed in reviews.

Experimental results on the Amazon "Electronic" and "Book" datasets demonstrate that the MENN model consistently outperforms baseline models such as FM, DNN, and DeepFM across all evaluation metrics (MAE, MSE, and RMSE). The findings verify that fusing numerical and textual information through the MENN framework significantly enhances prediction accuracy and model stability.

In conclusion, this research provides a practical and effective approach for multi-source user preference mining, contributing to the advancement of intelligent recommendation systems. Future work will focus on expanding the model's applicability to larger and more diverse datasets, optimizing the fusion mechanism for different modalities, and exploring the incorporation of contextual and temporal dynamics to further improve recommendation accuracy and personalization.

Acknowledgements

The author would like to thank all those who provided helpful feedback and encouragement during the completion of this work.

References

[1] Gao, M.; Li, J.Y.; Chen, C.H.; Li, Y.; Zhang, J.; Zhan, Z.H. Enhanced multi-task learning and knowledge graph-based recommen der system. IEEE Trans. Knowl. Data Eng. 2023, 35, 10281–10294.

[2] Cui, Y.; Yu, H.; Guo, X.; Cao, H.; Wang, L. RAKCR: Reviews sentiment-aware based knowledge graph convolutional networks for Personalized Recommendation. Expert

Syst. Appl. 2024, 248, 123403.

[3] Cheng, S.H.; Chen, W.Y.; Liu, W.L.; Zhou, L.; Zhao, H.L.; Kong, W.S.; Qu, H.; Fu, M.S. Dynamic training for handling textual label noise. Appl. Intell. 2024, 54, 11161–11176. Wit, E. and McClure, J. (2004) Statistics for Microarrays: Design, Analysis, and Inference. 5th Edition, John Wiley & Sons Ltd., Chichester.

[4] Kuo, R.J.; Li, S.S. Applying particle swarm optimization algorithm-based collaborative filtering recommender system consid-ering rating and review. Appl. Soft Comput. 2023, 135, 110038.

[5] Park, J.; Ahn, H.; Kim, D.; Park, E. GNN-IR: Examining graph neural networks for influencer recommendations in social media marketing. J. Retail. Consum. Serv. 2024, 78, 103075.

[6] Liu,W.; Zhang, Z.; Bai, Y.; Liu, Y.; Yang, A.; Li, J. A DeepFM-based non-parametric model enabled big data platform for predicting passenger car sales in sustainable way. IEEE Trans. Intell. Transp. Syst. 2023, 24, 16018–16028.