

RESEARCH ARTICLE



Personalized Recommender System for Children's Book Recommendation with a Real-Time Interactive Robot

Yun Liu¹, Tianmeng Gao¹, Baolin Song¹ and Chengwei Huang^{1,*}

¹Fandou Information Technology Co. Ltd., China

Abstract: In this paper we study the personalized book recommender system in a child–robot interactive environment. Firstly, we propose a novel text search algorithm using an inverse filtering mechanism that improves the efficiency. Secondly, we propose a user interest prediction method based on the Bayesian network and a novel feedback mechanism. The feedback is processed by converting the user's reaction into accept or reject. According to children's fuzzy language input, the proposed method gives the predicted interests, which is based on a common set of interests related to children's literature. Thirdly, the domain specific synonym association is proposed based on word vectorization, in order to improve the understanding of user intention. Additional phrases are extended to broaden the user's search range. Experimental results show that the proposed recommender system has an improved performance compared with the baseline results. Due to the algorithm efficiency design, it can operate on embedded consumer devices with limited computational resources.

Keywords: personalized search, word vectorization, recommender system, children's robot

1. Introduction

Recommender systems are widely used to recommend products and services to end consumers. Various studies have been reported on building a recommender system. It is a growing field attracting more and more researchers from various backgrounds.

Recommender system is very important in consumer electronics. Cheung et al. [1] used comments from an online Chinese discussion forum to evaluate the credibility of recommendations, as an extension for conventional word-of-mouth study. In their work, how readers would be influenced by online recommendations was studied. The quality of recommendation played an important role. Chen et al. [2] from Carnegie Mellon University also presented similar influence of online recommendations on consumer sales. Their work showed that recommender systems could improve sales at Amazon.com. User comments can be collected and used for recommendation by text mining techniques [3, 4]. Kumar and Benbasat [5] studied business-to-consumer website, and recommender system had improved both perceived usefulness and social presence. Lee et al. [6] studied program recommendations in Digital Television. Refining the TV program, selecting and satisfying the consumer's viewing requirements had a lot similarity with children's book recommendation. Both require adaptation and real-time response. In their work, many factors were considered to make good recommendation, such as viewing pattern for contents, statistical information of viewing patterns, and user's profile.

A novel approach to recommend music to consumers has been proposed by Ayata et al. [7]. User emotions were detected based on physiological signals and feeded to a recommendation engine for improved consumer satisfaction.

These works have already revealed the usefulness of recommender system in many aspects of consumer electronics, but the application in educational robots has not been fully studied, especially for the recommendation of children's books in a child–robot interactive environment. In our work, specific factors related to children's reading interests are analyzed and used for recommendation.

On the other hand, interactive AI has been drawing a lot of research attentions and it leads to many applications in intelligent toys. Park et al. [8] investigated interactive AI for education purposes. In their work, virtual environment was created for language learning. Various efforts have been made for children's education and companion through interactive AI. Ahn et al. [9] introduced an interactive toy for children to learn language. Their system used image recognition technology to response with the name of an object. It was also running on embedded system, but it was not connected with adequate online resources to facilitate the learning. Scheidt and Pulver [10] from Germany proposed a novel cube-like intelligent toy. It could be used for STEM teaching purposes. Sensors were available with trainable functionality for image classification, which was implemented on edge devices. However, the interaction mainly took place between machines, without much high-level communication with children. Faria et al. [11] proposed to use interactive games to assist the monitoring of children's brain activities. The interactive environment helped to elicit specific cognitive process.

*Corresponding author: Chengwei Huang, Fandou Information Technology Co. Ltd., China. Email: huangcw@fandoutech.com.cn

Natural language processing (NLP) plays an important role in interactive AI. Patel et al. [12] presented a chatbot for college students. Through natural language interaction, the robot could provide useful information to anyone who accessed the university website. Stefnisson and Thue [13] proposed a novel application in storytelling. Using NLP modules, the robot could help authoring the story contents. However, the AI agent could only understand simple relations and synonym association was not considered to improve the language understanding. Ping et al. [14] studied conversational AI during live conversation interactions. Cognitive principles were considered, which brought more adaptive and personalized language understanding ability. However deeper meaning of the text was not explored, such as user intention and user interests.

In our work, recommender system is designed in an interactive environment, in which children can talk freely with an intelligent robot and be guided to suitable reading and listening materials. Our robot is connected with a large amount of online books and resources, so that it may provide high quality content for children's education. Furthermore, the user intention is considered for better recommendation.

The recommender system can be viewed as a search ranking system, where the input query is a set of user and contextual information, and the output is a ranked list of items [15]. Memorization and generalization are two different techniques within a recommender system. Memorization learns the frequently co-occurred factors in historical data and infers the same response at the same input. Generalization uses machine learning algorithms to learn the patterns in the historical data so that it can infer new responses at similar inputs. For instance, Cheng et al. [15] proposed a framework called wide and deep learning combined a linear model component and a neural network component. It has good results on Google Play for recommending apps to users using massive-scale commercial app data. Another popular algorithm in recommender systems is collaborative filtering [16]. It can be coupled with deep learning for better content information ranking [17].

Although there are a few personalized search engines reported in the literature, the understanding of user intention remains a challenging problem. The following drawbacks in the conventional text search engine make it difficult to meet the actual needs of users.

First, computational cost of large-scale text search is still high. With the rapid increase of online database, the computational burden on embedded systems is too high. When a stable wireless connection is not available, the computation costs of search algorithms are dependent on a local mobile device. In order to improve the quality and the speed of search engine, the efficiency of the conventional search algorithms need to be further improved.

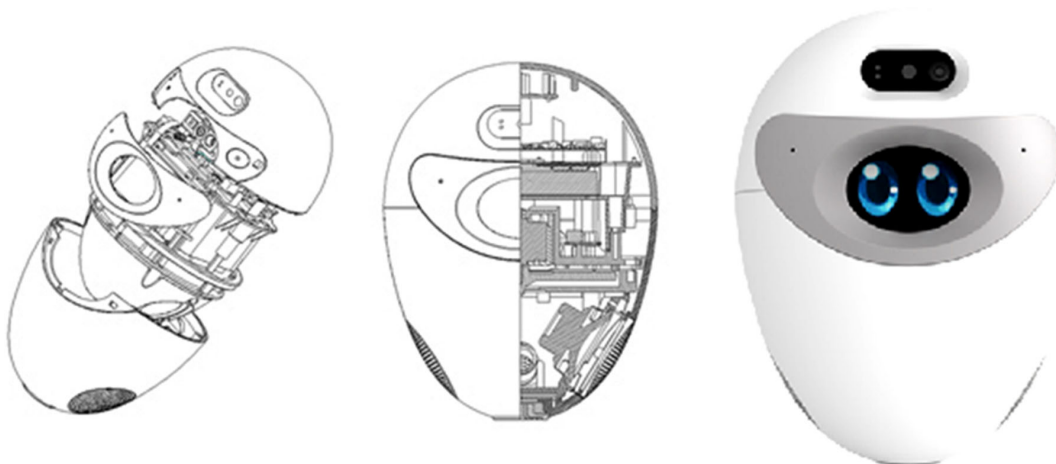
Second, user interests are not fully considered in the search process. The keyword-based text matching algorithms may not fully understand the user intention. Individual user's characters, personalities, search context, and interests should be considered in the search process. Searching is a passive way of recommendation; the matched text results should meet the actual needs of each individual user.

Third, semantic meaning of the large text data is not well represented in search algorithms. The text matching methods based on keywords are often lack of semantic understanding. It is difficult for the users to present every possible expression with the same meaning that they try to search in the text database. The recent developments in deep learning have resulted in an efficient representation of semantic meaning in words and phrases using word vectorization technique. When the search input is ambiguous and misleading, the search process becomes more difficult. Word vectorization may help to associate different keywords for matching. On the other hand, using natural language descriptions is more convenient than separate keywords and provides better user experience.

A word embedding method based on the deep learning is proposed by Mikolov et al. [18] and Mikolov et al. [19] named word2vec. While applying the neural network method to the conventional natural language model, the words can be represented as vectors in high-dimensional space. The word embedding model is learned on a large text database. The relation between two words can be represented as the geometric distance between vectors. This provides a new perspective for text search systems. The difficulties in semantic understanding can be efficiently addressed by vectorization process. The keywords in search process can be associated with a list of related keywords that broaden the search range and retrieve deeper information from text databases.

In this paper, we study the personalized recommender system with a real-time interactive robot for children, as shown in Figure 1. The robot can interact with users through speech and

Figure 1
The children's interactive robot



vision. A child user can tell the robot to start a reading session and what kind of books he or she is interested in. The robot needs to infer from the context and recommend books based on a child's words, which are usually ambiguous and inaccurate. The visual recognition ability enables the robot to understand the book content and tell a story [20].

We propose a novel text search algorithm using an inverse filtering (IF) mechanism that is very efficient for labeled items search. The Bayesian network is used to implement the user interest prediction (IP) for an improved personalized search. It not only searches the related items using keywords but also predicts user interests. The word vectorization is used during the process to discover potential targets with similar semantic meanings. The main contributions of our proposed system are three-fold: (i) an efficient search algorithm based on IF is proposed; (ii) a user IP method based on Bayesian network is implemented and combined with a personalized feedback mechanism; and (iii) based on the word vectorization technique, a domain specific synonym association (DSSA) method is proposed to calculate the word distance in specific knowledge domain.

Key functionalities of our robot include, but not limited to, book reading, conversation, and stories telling. Recommender system is one of the most important gateways for child users to learn with our robot. It separates this robot apart from other non-intelligent learning tools.

Since 2015, this robot, along with the proposed recommender system, has helped more than 10,000 homes in the consumer market and expected to continue growing. The algorithms proposed in this paper are implemented in the embedded systems of our robots, which are deployed in real-world environment satisfying real-time interaction demands.

The rest of the paper is organized as follows: Section 2 provides related work on personalized recommender systems; Section 3 gives the baseline methodology on fuzzy search and IP, as well as the overall system architecture; Section 4 introduces the details of DSSA method; experimental results are shown in Section 5; finally, conclusions are drawn in Section 6.

2. Related Work

During the real-time interaction between a robot and a user, keywords can be used as an important clue for searching and recommendation. Full-text search engine is based on keyword matching technology, which can be used in real-time recommendation. Fuzzy search methods, KMP [21] and BM [22], are proposed for approximate fast text matching.

However, a full-text search engine cannot understand the user's deep semantics and needs, especially for fuzzy natural language input by the user. The traditional search engine returns the same result when different users enter the same query. Its search results are often inconsistent with the user's needs.

The personalized recommendation refers to the use of scenes, time, and other information to understand user's real intention. The understanding of the natural language text is consistent with people's everyday thinking; therefore, it can be used to build a smarter personalized recommender system [23]. For instance, the natural language search engine Ask Jeeves supports for natural language search. Users could retrieve information through Q&A because there was a huge database of problems in the backend.

Several existing works studied book recommendation [24–26]. Tewari et al. [26] investigated various approaches for book

recommendation, including content filtering, collaborative filtering, and rule-based recommendation. Collaborative filtering [27] is widely used in the existing online retail providers. In this algorithm, the users are given recommendations based on the similarity between its profile and other users' profiles. The similarity is defined on various metrics. Rana and Jain [25] looked into the application in online book shopping. In their work, a method with diverse recommendation was studied, and the recommended list might change over time. They adopted a temporal dimension that measures how much an item was liked by the user, which was time variant. Hariadi and Nurjanah [24] studied the personality in book recommendation. The personality factor in their work was built according to the similarity between users. The relation between user profile and user interest need to be further studied.

3. Personalized Recommender System for Children's Robot

We study the search algorithm in the context of children's book recommendation. The books in the database are associated with text labels, such as names, authors, topics, and other keywords. Matching these labels can be a straightforward way to search books. Mining the potential relations between these labels and a child's interest can provide better recommendation results.

3.1. Robot architecture

The main chip has a four-core CPU whose frequency is up to 1.2GHz and a GPU that supports OpenGL. The main chip is connected with 1GB memory and 8GB flash drive space. The camera supports 5M pixel format, and it is responsible for the visual interaction function. The audio codec supports 2-way speaker, 4-way microphone, and 4-way earphone. They are responsible for the voice interaction function. The MCU is connected to the main chip with I2C, UART, and GPIO ports. It is responsible for motion control. The overall system power is 10W.

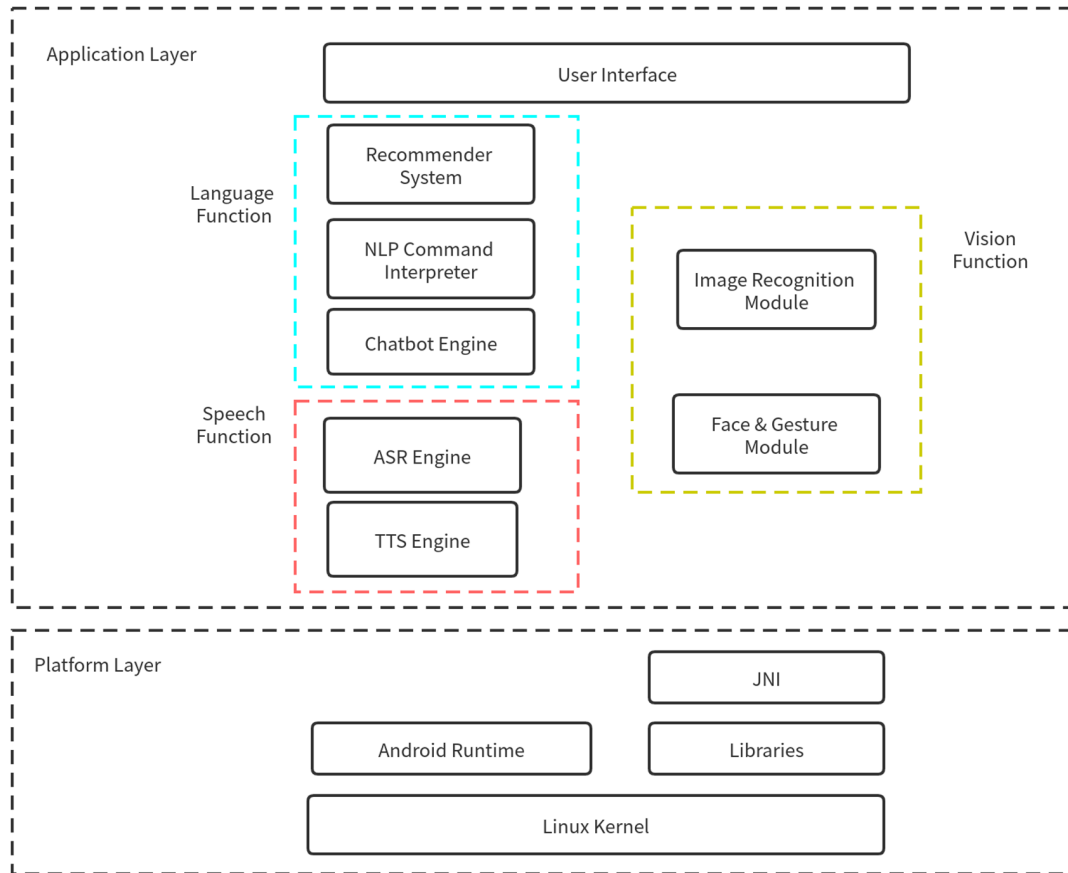
The software architecture of our robot is shown in Figure 2. The overall architecture consists of two layers, the platform layer and the application layer. In the application layer, we develop key functions including visual interaction, speech interaction, and language understanding. The visual interaction is introduced in our previous work [20]. The speech interaction function involves real-time signal processing and enhancement. It is a natural and important mean to communicate with human [28]. Different noise backgrounds are first detected, and speech end point segmentation is carried out for preprocessing. Finally, the text content of the speech signal is generated using a local Automatic Speech Recognition (ASR) engine or a cloud-based ASR service, depending on the network connection quality.

Children's picture books are recognized and read to children. Before the robot can tell a story about any book, the recommender system is run and proposes books of interest. The recommender system is built together with other language modules, such as chatbot module and command understanding module. In the platform layer, JNI (Java Native Interface) is used to compile C++ libraries for faster computing.

3.2. Approximate string match using IF

Various search algorithms can be used for text matching [29, 30], but text matching can be time consuming. In this section, we propose to use a method called IF to accelerate the search

Figure 2
The SW architecture of our interactive robot



process. The philosophy behind this method is based on two parts, the inverse part and the filtering part. The inverse idea refers to that, instead of going through a large amount of text to find a match with any keyword in the conventional approximate string match methods, we go through a small list of user input text contains some keywords to find matches with a large amount of book labels in the database. The filtering idea refers to that we extract a very small alphabet from the user input text and use it to filter out any label that does not contain the characters in the alphabet. A predefined hash data structure makes this filter process very fast and the alphabet extraction works very well for Chinese language which is based on a large number of characters. Algorithm details will be given in the remaining of this section.

The inverse and filtering idea has been adopted previously in acoustic signal processing [31].

The IF-based search algorithm mainly includes three steps as described below.

Step one: Building the hash index of resource database.

Before we search any children book, we label them using names, authors, topics, etc. Based on the book content, we extract representative labels automatically using Term Frequency–Inverse Document Frequency (TF-IDF) algorithm [32, 33]. In practice, we manually reviewed these labels on approximately 3126 books for a better quality. TF-IDF algorithm is used to determine the weight of every keyword in the document, and the high ranking words are considered as the representative labels. In this paper, the phrases book labels and book keywords refer to the same

meaning, and label and keyword may be used interchangeably in the rest of the paper.

One fact about the Chinese language is that the number of characters in commonly used words and sentences is limited to a small number, approximately independent of the size of the book database. The number of characters tends to be stable when the number of keywords increases. For a Chinese database containing about 1.0×10^4 keywords, the number of characters is only around 3.5×10^2 . So a hash map can be used to filter out the irrelevant keywords quickly.

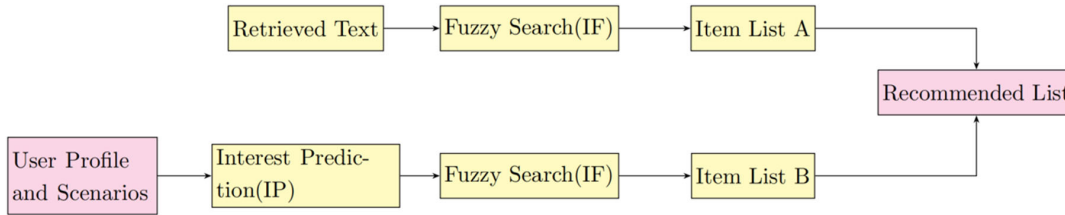
Due to this property, the hash map is establishment beforehand. A character may be contained by a keyword, and a keyword may be contained by a book. Therefore, the first hash map D1 is a map from any one character to a list of keywords that contains it. The second hash map D2 is a map from any one keyword to a list of books or items in the database that contains it.

Step two: Match keywords using IF. The process of IF algorithm can be divided into two sub-steps.

First, go through the input text X entered by the user in one search action and for each character C in X use the hash map D1 to pick out the related keywords to build a candidate set L_{cand} . This process lies in the heart of IF, and it accelerates the search process significantly.

Second, go through all the keywords in L_{cand} and for each keyword using a sliding window on the input text X to find an approximately matched string if it exist. Dynamic time wrapping is used to compare sub-strings with different length and calculate

Figure 3
The baseline framework of personalized fuzzy search system



the similarity scores. If the matching score is above the threshold, the corresponding keyword is put in the hit label set L_{hit} .

Step three: Post-process for ranking results.

When the input text provided by the user contains the exact expression of the targets in the database, the matched results are accurate and usually satisfactory. When the input text is ambiguous or misleading, a series of fuzzy results will be bit by the IF search algorithm. A post-process for ranking the similarity score in the hit set L_{hit} is necessary. The definition of the similarity score is shown in Equation (1), index $i = 1, 2, \dots, N$ is the number of the keyword, and the distance between the keyword and a string contained in the sliding window is denoted as $dist_{DTW}(Key(i,m),X)$. The index m refers to different books or items in the database.

$$score(m) = \sum_{i=1}^N \frac{length(Key(i, m))}{dist_{DTW}(Key(i, m), X)} \quad (1)$$

Using hash map D2, we can find the books related to any keyword. When a keyword is matched, we can find out which book it belongs to. The results are ranked according to the scores and returned to the user.

3.3. IF-based fuzzy search framework

In this section, we introduce the basic configuration of our recommender system. It consists of user IP module, IF search module, and a feedback mechanism. The extension of the basic engine with a word association module is introduced in Section 4. The overall system flowchart is shown in Figure 3. The design of this engine is suitable for various general purposes, but it has a special focus on the recommender system in a virtual library where users share their opinions and comments on children’s books.

The IF-based keyword search module works independently at first taking in input sentences explicitly said by a user. The search result is denoted as L_{exp} , which consists of a list of book names (items). The user IP module takes in several factors from the user profile, such as age, gender, and hobbies. It also takes some environmental factors as the input, such as time, date, and interactive state. Our search engine is deployed on an intelligent robot for dialogue-based book recommendation. Therefore, the interactive states may be important to predict user’s search intention, such as playing a game and learning English. The predicted user interests are then converted into a string of keywords, and IF search algorithm is applied once more to generate another search result denoted as L_{exp} . The final search result is a combination of these two lists. A set of empirical rules is used to merge the two lists: (a) if list L_{exp} contains high score matches, they should have higher ranks; (b) the common item from list L_{IF} and L_{pre} should have higher ranks; and (c) the sizes

of the lists should be limited. We take a direct implementation of these rules in simple IF-THEN form.

The user IP can be very helpful. For instance, the input of “I am thirsty” from a user might trigger a response with “football,” “drinks,” “sports bar,” etc. in the interest list L_{pre} and “sports bar,” “banana milk,” etc. in the L_{exp} list. A combination of the interests and match results will place “sports bar” at the front of the search result list.

Consider some human–machine interaction application, the user might provide useful feedbacks on the search results. These feedbacks can be converted into simple accept (success) or reject (failure) in binary forms. A feedback adaptive mechanism is proposed to improve the IP module for different individual user. The longer the prediction module is used, the better it is adapted and personalized for the user.

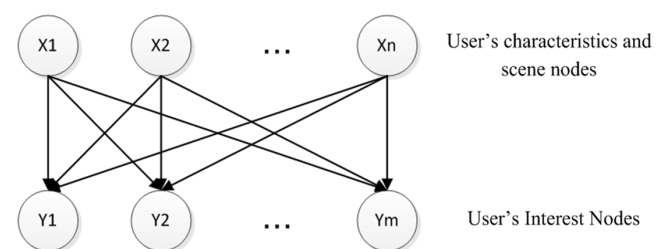
3.4. User IP with feedback mechanism

In this section, we implement the user IP module as described above using Bayesian network [34, 35]. The directed graphical model is used for inference user interest based on related factors like age, gender, personalities, hobbies, environmental factors, scenes, etc.

The network model is demonstrated in Figure 4, input layer nodes denote the observed variables (user profile factors), and they are independent of each other: $X = \{X_1, X_2, \dots, X_n\}$. X_i represents a keyword describing user’s age range, gender type, personality type, hobby, time, day, scene, or any related environmental factor. The predicted interests are represented as language variables: $Y = \{Y_1, Y_2, \dots, Y_m\}$. Each keyword is dependent on the combination of all the input variables. Through a training process, the conditional probability distribution is learned.

Denote the conditional probability distributions in the network as: $\theta = \{P(Y_j|X_i)\}$. Maximum a posteriori algorithm is used to estimate the parameters [36]. The node in the output layer is dependent on the parent nodes in the above layer.

Figure 4
The network structure of user interest prediction



Given $Y_j = k$, the posteriori probability is calculated as:

$$P(X_i = h | Y_j = k) = \frac{\sum_{t \in \Phi} I(Y_j^{(t)} = k) \sum_{q \in \Omega} I(X_i^{(q)} = h)}{\sum_{t \in \Phi} I(Y_j^{(t)} = k) \times n_t} \quad (2)$$

where k denotes the interest index, h denotes the factors in user profile, t is an index of training samples, Φ is the set of training samples, Ω is the set of user profile factors, and q is the index of the variables in the user profile set. $I()$ is an index function, and if the expression is true, it takes the value 1, otherwise it equals to 0.

In the above-described Bayesian network-based user IP, an adaptive feedback adjustment mechanism is used to incrementally retrain the Bayesian network model. The longer a user uses this engine and gives feedback on the search results, the better it is personalized to this user. Feedback on success prediction will enhance the connection between the current inputs and outputs in the network; on the other hand, feedback on false search results will punish the prediction.

User's feedback is presented in the binary form:

$$S = \begin{cases} 0 & \text{reject} \\ 1 & \text{accept} \end{cases} \quad (3)$$

The feedback mechanism of network parameters is based on the evaluation of the user's prediction results. The probability distribution $\theta = P(Y_j | X_i)$ is updated in the current network model. d_{new} is the incremental training data corresponded to the user feedback. It consists of the observable node set $cX_{observed}$, the predicted interest node Y_j , and the weight set W between these probabilistic nodes. We have

$$d_{new} = \{ (X_1, X_2, \dots, X_l, Y_j, W_{1j}, W_{2j}, \dots, W_{lj}) | (X_1, X_2, \dots, X_l) \in X_{observed}, Y_j \in Y_R, (W_{1j}, W_{2j}, \dots, W_{lj}) \in W \} \quad (4)$$

where Y_R is the set of output variable. We update the weights as shown in Algorithm 1.

Algorithm 1. The incremental learning process of proposed feedback mechanism

```

Input:  $W_{old}$ 
Output:  $W_{new}$ 
1: Initialize:  $W_{new} \leftarrow W_{old}$ 
2: for  $X_i \in X_{observed}$  do
3:   weight  $\xrightarrow{W_{ij} \in W_{old}}$   $W_{ij}$ 
4:   if  $S == 1$ 
5:     weight = weight + th1
6:   else
7:     weight = weight - th2
8:   end if
9:    $W_{ij-new} \leftarrow W_{ij-new} \cup W_{new}$  weight
10: end for
    
```

4. DSSA for Improved Recommendation

In order to understand the text semantic meaning in user input, word vectorization is used as a plug-in module to improve

the above-mentioned IP and IF search. Specifically, a DSSA method is used for user preference setting.

4.1. Domain-specific synonym association

IP may extend the search range on children book labels. However, some of the interest keywords are not directly linked to the labels if we do not understand the semantic meanings. Therefore, in this section, we use the word vectors to associate one keywords to several of its nearest neighbors in the vector space. This process helps to understand the user's intention and expansions of the search results.

A direct application of word2vec tool is able to associate some words to extend the search range, but we found that the neighboring words in the vectors space are not always contained in children books. Therefore, a domain-specific word association method is needed to locate words within special topics.

The conventional word2vec model is retrained on a database of children books to make sure that the labels are represented in this vector space. Suppose there are N words in total, and they are represented as vectors: $W = \{w_1, w_2, \dots, w_N\}$.

The book labels are also represented as K vectors: $C = \{c_1, c_2, \dots, c_K\}$. A spatial partitioning can be drawn with minimum mean distortion to classify w_i to c_j . Hence, we can map the unrestricted words to a predefined set of book labels C .

Denote the dimension of a word vector is m . A word can be represented as $w_n = (w_{n1}, w_{n2}, \dots, w_{nm})$, where $1 \leq n \leq N$. A target word (book label) can be denoted as $c_k = (c_{k1}, c_{k2}, \dots, c_{km})$, where $1 \leq k \leq K$.

For each vector c_k in the target word set C , there is only one spatial region S_k corresponded to it. The target word vector and the division of the K space are one to one mapping. Denote spatial regions as $P = \{S_1, S_2, \dots, S_K\}$, where S_k corresponds to vector c_k . When a word vector w_n is within the region of S_k , the mapping from the word vector w_n to word vector c_k can be obtained from Equation (5).

The mean square error can be used to measure the distortion, and it is defined as:

$$f : W \rightarrow C \quad (5)$$

$$f(w_n) = c_k, w_n \in S_k$$

The mean square error can be used to measure the distortion, and it is defined as:

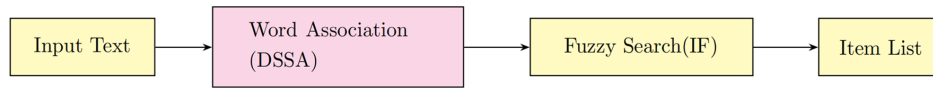
$$\sigma_{ave} = \frac{1}{N} \sum_{n=1}^N \|w_n - f(w_n)\|^2 \quad (6)$$

In order to find the optimal set $P = \{S_1, S_2, \dots, S_K\}$ that minimizes the average error σ_{ave} , P needs to satisfy the nearest neighbor condition, that is, the region S_k should contain all the words closest to the vector c_k :

$$S_k = \{w | \|w - c_k\|^2 \leq \|w - c_{k'}\|^2, \forall k' = 1, 2, \dots, K\} \quad (7)$$

This reduction occurs because domain-specific sets focus on terms that are specifically relevant to a particular field or subject, filtering out more general or unrelated terms. It further improves the computational efficiency. The resulting associated synonym falls into the book labels set and the search using that synonym is more likely to hit a target book.

Figure 5
Improved IF module with domain-specific synonym association



4.2. Overall architecture

Using the DSSA, the extended recommender engine with word2vec plug-in is depicted in Figure 5.

Both the user input words and predicted interest keywords are extended by word2vec module, before sending to IF-based search module. The predicted interest descriptions are converted into keywords from the book labels in the database for an improved search experience, which will be demonstrated in Section 5.

Using a retrained word2vec module as introduced in Section 4.1, the keywords of the IF search module will be extended according to the semantic meaning of the input text. The associated keywords that generated by the word2vec module are based on the domain-specific vocabulary of stories from children books. This plug-in module makes the search system more intelligent and more adaptive to children’s needs.

5. Experimental Results

In our experiments, we verify the performance of our proposed recommender system in the following four aspects. First, Section 5.2 shows the results using only IF module. Second, Section 5.1 shows the performance of our proposed novel IP module. Third, Section 5.3 presents the baseline recommender system using IF module and IP module. Finally, in Section 5.4, using the proposed novel DSSA module, the performance is compared with the conventional ones.

The following experimental results are tested on a PC with simulated environments for the convenience of reproducibility, except the embedded testing results in Table 1. Nevertheless, all the data sets used are recorded from real-time interaction with children. All the algorithms are implemented on our embedded robot system.

5.1. IF-based fuzzy search: Accuracy and efficiency

The experiments on fuzzy search methods, KMP and BM, are adopted for comparison with the proposed IF-based method. The algorithms are implemented using Java. The experiments on PC are carried out with 3.20 GHz CPU and 8GB memory. The tests on robot are carried out with 1.2 GHz CPU and 1GB memory. The NLP toolkit FudanNLP [37] is used to assist the extraction of keywords from each book’s content.

Table 1
Comparison of three fuzzy search algorithms

Fuzzy search algorithms	Accuracy	Response time (PC)	Response time (robot)
KMP	89.29%	4186 ms	1038 ms
BM	89.29%	1679 ms	3977 ms
IF	91.27%	607 ms	1521 ms

The search results are presented in book names. A search is considered successful when the result book names contain the predefined book (ground truth) that the user actually wants to find with a certain inquiry text (usually a sentence, or several sentences) he or she provided. The number of the book names returned by the search engine affects the success rate. The more books are allowed to present the higher the success rate goes, and the less smart the engine looks. In extreme condition, if we return all books in the database as the search result, the rate reaches 100%, suppose the targets are all included in the database. The IF-based search, KMP matching algorithm, and BM matching algorithm are compared in keyword search. The experimental results, tested both on PC and on our robot, are shown in Table 1. The success of our keyword search test is defined differently from the above-described book search. When a user provides an inquiry sentence that contains a keyword in an inexact form (missing a character or a wrong character), the search algorithms return a number of keywords matched. If the inquiry keyword is found among the returned keywords, it is considered as a success.

We can see from Table 1 that KMP-based search and BM-based search are the same in accuracy rates. The accuracy of the proposed IF algorithm is higher than the other two conventional algorithms. The response time of KMP is the longest, and the IF method is significantly faster according to the tested response time. The efficiency of our search engine is improved by IF due to its hash map removing irrelevant candidate keywords.

5.2. User IP results

In order to verify the effectiveness of the user IP method based on Bayesian network, we designed a test set that is consisted of two parts, the user profile and the user interest. The prediction problem is to find a map from user profile variables to user interest variables. Subjects are required to select 5–8 keywords that fit his or her profile and select 10 keywords from the a label set describing the common children’s interests, for example, football, video game, apple juice, reading, etc. This type of predicted interest keywords may be considered as individual’s intention during vocal interaction with a robot. We may infer the proper response based on the predicted keyword. We use precision rate and recall rate [38] to evaluate the performance of the prediction algorithm. K-fold cross-validation is used to prevent over-fitting.

As shown in Figures 6 and 7, the prediction results are represented as a number of interest keywords, and as the number of predicted keywords increases, the precision rate and the recall rate vary. The precision decreases when the number of predicted keywords increases, because some irrelevant words are introduced to the results. The recall rate increases when the number of predicted keywords increases, because more interest keywords are returned and it is less likely to miss the targets.

The training size to test size ratio starts from 70%, as we increase the training samples from user feedback and retrain the model incrementally, the performance is improved as observed in both precision rate and recall rate. Therefore, the proposed feedback mechanism is effective.

Figure 6
The precision rate of user interest prediction

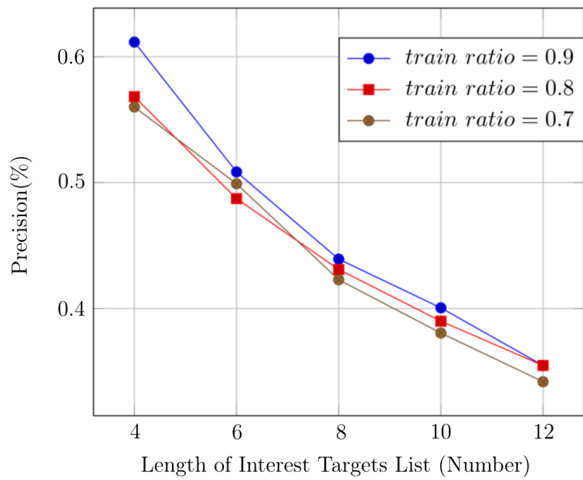
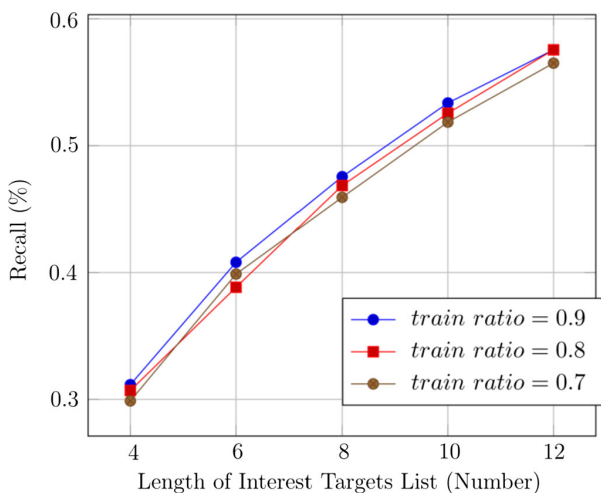


Figure 7
The recall rate of user interest prediction



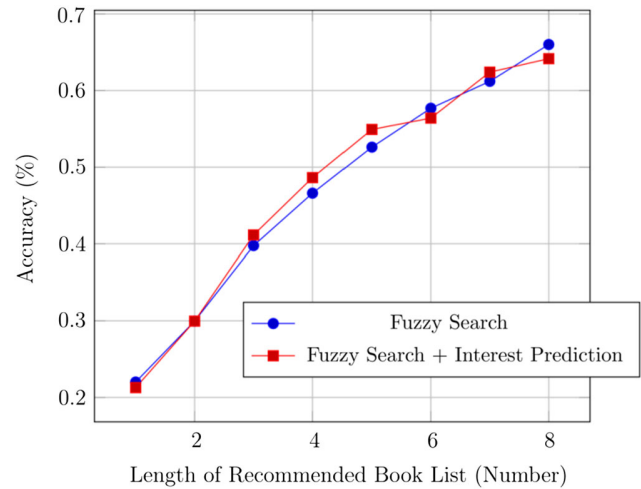
5.3. Personalized recommender system: Overall performance

In this experiment, we test the children book searching with different user profile setting. The user prediction and IF-based search are integrated to perform the personalized fuzzy search. In the test, we select 500 samples from 100 subjects. Those test samples contain the description of what the user wants to search for as the input text, the user profile described in separate keywords form, and the ground truth book name(s) the user actually wants to find as the target.

As show in Figure 8, the search accuracy is slightly improved in most of the settings, by using the user IP method. When the number of returned book in the search result is set to 4 or 5, the improvements are more obvious.

Some examples in this search experiment are given in Table 2. We can see that the search system can recommend books suitable for different users. This search results are therefore more personalized.

Figure 8
Search result with or without user interest prediction



The reason behind the user IP is that more keywords could be used in searching just knowing who the user is from the user profile without explicitly asking what the user what he or she wants to search.

However, the problem of using this type of predicted interest labels is that the words and phrases we choose to describe a user’s interests are not the same as what we choose to describe a book. Therefore, the search result in Figure 8 is not improved significantly. For instance, we may describe a user as “fond of ping-pang,” when he may be looking for a book named “fun games after school.”

A conventional way to bridge this gap is to use the same dictionary (a common set of words and phrases) when we label the books and setting user profiles. This may improve the search results directly. However, in this fashion, whenever we want to apply the recommender engine to a different database or a different group of users, we need to change the dictionary and rebuilt every data hash. In this paper, as we proposed in Section 4, we plug-in word2vec module to overcome the difficulty in semantic understanding between “fond of ping-pang” and “fun games after school” in an efficient way.

5.4. Improved personalized recommender system: DSSA

In this section, we test the performance using word association. The training corpus used for word2vec module is from the news corpus from Sogou laboratory [39] including 18 channels of news, e.g., domestic news, international news, sports news, and entertainment news, recorded between June and July, 2015. When we apply the word association directly using this word2vec model, the results are shown in Table 3.

It is obvious that the results are dependent on the content of the training corpus. The domain-specific association can bring the results closer to book labels in the database. The final search performances are compared in Figure 9. Using synonym association, we may provide more diversified response adapted to user intention, since the DSSA is based on the book content and user input.

We can see from Figure 9 that the basic synonym association module does not improve the result. It associates the keywords to

Table 2
Examples of personalized search results (translated in English)

Input inquiry text	User profile	Predicted interest keywords	Search result (book)
“Stories Related to Tom”	Girl, night	Sleep, moon, reading, piano	Tom’s Nightmare
“Stories Related to Tom”	Boy, summer	Game, swimming, soccer, basketball	Tom Is in The Swimming Pool
“Stories Related to Emma”	Boy, winter	Game, soccer, snowman, basketball	Emma and Snowball Fight
“Stories Related to Emma”	Girl, extroverted	Reading, sleeping, communication, laughing	Emma and Her Friends

Table 3
The results of word vector association (translated in English)

Words	Word2vec associative synonyms	Word2vec directional association synonym
Musicmythology	Play, singing, zhongshan, diction	Play, sing, organ, poetry
Sun	Hebrew, orphans, Gang, Greek other side, the past, 4 pieces, Moses	Greece, Paris, history, astronomer
Butterfly	Primrose, intake, half, yellow	Shine, go far, east, sky yellow, chrysanthemum, orchid, pink

6. Conclusions

In this paper, we study the personalized text search problem. The keyword-based search method in conventional algorithms has a low efficiency in understanding users’ intention since the semantic meaning, user profile, and user interests are not always considered. We propose a novel text search algorithm using an IF mechanism that is very efficient for label-based item search. Furthermore, we adopt the Bayesian network to implement the user IP for an improved personalized search. According to user input, it searches the related items using keyword information and predicted user interest. In our approach, the word vectorization is used to discover potential targets according to the semantic meaning. Results show that the proposed recommender system has an improved efficiency and accuracy, and it can operate on embedded consumer devices with very limited computational resources.

There are still several limitations in our proposed recommender system. Our search methods are only based on the short-time interaction containing few rounds. Deeper understanding of multi-round interaction may improve the ability of intention prediction. The content we used for recommendation is limited to children’s books. Extension to general knowledge and other learning materials using knowledge graph may be a valuable direction to study. Recommending the learning path and personalized learning plan is also out of our scope. In our future work, we will further investigate large language models to understand multi-round, multi-channel interaction where context information, facial expression, gestures can be considered for intention prediction.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

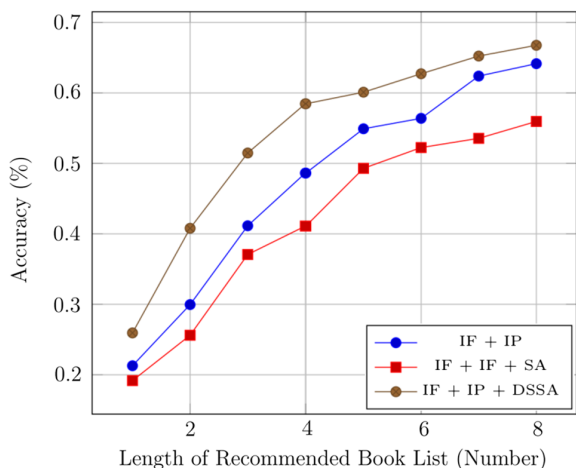
The data that support the findings of this study are not publicly available due to involvement with commercial products.

Author Contribution Statement

Yun Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Tianmeng Gao:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Baolin Song:** Conceptualization, Methodology, Software, Validation,

Figure 9

Compare the final search results using improved DSSA module



semantically close words and phrases in an open domain. The book labels are not more likely to be hit than any other words in a large corpus. By introducing more irrelevant keywords for searching, the result accuracy decreases.

The use of DSSA improved the final results significantly. The predicted words from user IP and the associated synonyms are mapped onto the book labels in the semantic distance space created by word vectorization. Therefore, the recommendation results are improved.

Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Chengwei Huang:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

References

- [1] Cheung, M. Y., Luo, C., Sia, C. L., & Chen, H. (2009). Credibility of electronic word-of-mouth: Informational and normative determinants of on-line consumer recommendations. *International Journal of Electronic Commerce*, 13(4), 9–38.
- [2] Chen, P. Y., Wu, S., & Yoon, J. (2004). The impact of online recommendations and consumer feedback on sales. In *Proceedings of International Conference on Information Systems*, 711–724.
- [3] Aciar, S., Zhang, D., Simoff, S., & Debenham, J. (2007). Informed recommender: Basing recommendations on consumer product reviews. *IEEE Intelligent Systems*, 22(3), 39–47.
- [4] Benlian, A., Titah, R., & Hess, T. (2012). Differential effects of provider recommendations and consumer reviews in e-commerce transactions: An experimental study. *Journal of Management Information Systems*, 29(1), 237–272.
- [5] Kumar, N., & Benbasat, I. (2006). Research note: The influence of recommendations and consumer reviews on evaluations of websites. *Information Systems Research*, 17(4), 425–439.
- [6] Lee, S., Lee, D., & Lee, S. (2010). Personalized DTV program recommendation system under a cloud computing environment. *IEEE Transactions on Consumer Electronics*, 56(2), 1034–1042.
- [7] Ayata, D., Yaslan, Y., & Kamasak, M. E. (2018). Emotion based music recommendation system using wearable physiological sensors. *IEEE Transactions on Consumer Electronics*, 64(2), 196–203.
- [8] Park, W., Park, D., Ahn, B., Kang, S., Kim, H., Kim, R., & Na, J. (2019). Interactive AI for linguistic education built on VR environment using user generated contents. In *Proceedings of the 21st International Conference on Advanced Communication Technology*, 385–389.
- [9] Ahn, J. Y., Kim, D. W., Lee, Y. H., Kim, W., Hong, J. K., Shim, Y., & Kim, S. W. (2018). MOYA: Interactive AI toy for children to develop their language skills. In *Proceedings of the 9th Augmented Human International Conference*, 36.
- [10] Scheidt, A., & Pulver, T. (2019). Any-cubes: A children’s toy for learning AI: Enhanced play with deep learning and MQTT. In *Proceedings of Mensch und Computer*, 893–895.
- [11] Faria, D. R., Bird, J. J., Daquana, C., Kobylarz, J., & Ayrosa, P. P. (2020). Towards AI-based interactive game intervention to monitor concentration levels in children with attention deficit. *International Journal of Information and Education Technology*, 10(9), 641–648.
- [12] Patel, N. P., Parikh, D. R., Patel, D. A., & Patel, R. R. (2019). AI and web-based human-like interactive university chatbot (UNIBOT). In *3rd International Conference on Electronics, Communication and Aerospace Technology*, 148–150.
- [13] Stefnisson, I. S., & Thue, D. (2018). Mimisbrunnur: AI-assisted authoring for interactive storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 14(1), 236–242.
- [14] Ping, Q., Niu, F., Thattai, G., Chengottusseriyil, J., Gao, Q., & Reganti, A. (2020). Interactive teaching for conversational AI. *arXiv Preprint:2012.00958*.
- [15] Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., . . . , Shah, H. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 7–10.
- [16] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 291–324). Springer.
- [17] Wang, H., Wang, N., & Yeung, D. Y. (2015). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining*, 1235–1244.
- [18] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint:1301.3781*.
- [19] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 26, 3111–3119.
- [20] Huang, C., & Jiang, H. (2019). Image indexing and content analysis in children’s picture books using a large-scale database. *Multimedia Tools and Applications*, 78(15), 20679–20695.
- [21] Knuth, D. E., Morris, J. H., & Pratt, V. R. (1977). Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2), 323–350.
- [22] Boyer, R. S., & Moore, J. S. (1977). A fast string searching algorithm. *Communications of the ACM*, 20(10), 762–772.
- [23] Guo, W., Gao, H., Shi, J., & Long, B. (2019). Deep natural language processing for search systems. In *Proceedings of the 42nd International ACM Conference on Research and Development in Information Retrieval*, 1405–1406.
- [24] Hariadi, A. I., & Nurjanah, D. (2017). Hybrid attribute and personality based recommender system for book recommendation. In *International Conference on Data and Software Engineering*, 1–5.
- [25] Rana, C., & Jain, S. K. (2012). Building a book recommender system using time based content filtering. *WSEAS Transactions on Computers*, 11(2), 27–33.
- [26] Tewari, A. S., Kumar, A., & Barman, A. G. (2014). Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining. In *IEEE International Advance Computing Conference*, 500–503.
- [27] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communication of the ACM*, 35(12), 61–70.
- [28] Huang, C., Song, B., & Zhao, L. (2016). Emotional speech feature normalization and recognition based on speaker-sensitive feature clustering. *International Journal of Speech Technology*, 19, 805–816.
- [29] Chen, Y., Wang, W., Liu, Z., & Lin, X. (2009). Keyword search on structured and semi-structured data. In *Proceedings of the 2009 ACM SIGMOD International Conference on the Management of Data*, 1005–1010.

- [30] Fu, Z., Ren, K., Shu, J., Sun, X., & Huang, F. (2016). Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Transactions on Parallel and Distributed Systems*, 27(9), 2546–2559.
- [31] Rothenberg, M. (1973). A new inverse-filtering technique for deriving the glottal air flow waveform during voicing. *The Journal of the Acoustical Society of America*, 53(6), 1632–1645.
- [32] Qaiser, S., & Ali, R. (2018). Text mining: Use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1), 25–29.
- [33] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523.
- [34] Atoui, M. A., Cohen, A., Verron, S., & Kobi, A. (2019). A single Bayesian network classifier for monitoring with unknown classes. *Engineering Applications of Artificial Intelligence*, 85, 681–690.
- [35] Pearl, J. (1985). Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society*, 329–334.
- [36] Bassett, R., & Deride, J. (2019). Maximum a posteriori estimators as a limit of Bayes estimators. *Mathematical Programming*, 174(1), 129–144.
- [37] Qiu, X., Zhang, Q., & Huang, X. (2013). FudanNLP: A toolkit for Chinese natural language processing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 49–54.
- [38] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, 14(2), 1137–1145.
- [39] Wang, C., Zhang, M., Ma, S., & Ru, L. (2008). Automatic online news issue construction in web environment. In *Proceedings of the 17th International Conference on World Wide Web*, 457–466.

How to Cite: Liu, Y., Gao, T., Song, B., & Huang, C. (2025). Personalized Recommender System for Children’s Book Recommendation with a Real-Time Interactive Robot. *Journal of Data Science and Intelligent Systems*, 3(2), 98–108. <https://doi.org/10.47852/bonviewJDSIS3202850>