# CHAPTER TWO

## LITERATURE REVIEW

### 2. 0 INTRODUCTION

The purpose of this literature review is to critically evaluate the existing research on deep learning models for personalized book recommendations. The aim is to identify the strengths and limitations of current approaches, highlight the research gaps and challenges and provide a basis for the development of a novel deep learning model for personalized book recommendation.

### 2.1 BOOK RECOMMENDATION SYSTEMS

Book recommendation systems are software applications that suggest books to users based on their interests and preferences. The goal of these systems is to help users discover books that they might be interested in reading but might not otherwise have discovered on their own.

There are two types of recommendation systems i.e., personalized and non-personalized recommendation system. Personalized recommendation systems recommend product or items to users based on preference criteria of individual (Sana, James & Nasseh, 2019). This system makes use of the individual past records of user’s ratings in order to find more suitable items that are yet to be rated by the user as recommendations (Khalid & Jamshed, 2020). On the other hand, non-personalized recommendation systems recommend base on the current popular trends that are relevant to all users. These trends can be the top 10 most popular items among users, information on a certain update or the current updates of a particular environment which are suitable for every new user (Khatwani & Chandak, 2016).

In recent years, deep learning models have become increasingly popular for book recommendations (Dhanashri et al., 2020). These models use artificial neural networks and other machine learning techniques to process large amounts of data and make recommendations based on patterns and trends in the data (Balaji, Pranshu & Deepali, 2020). Deep learning models can incorporate a wide range of data, including user behavior, book metadata, and social network information, to make highly personalized recommendations (Alexandros & Balázs, 2017).

However, there are several challenges associated with developing effective book recommendation systems. These include the cold start problem, which occurs when there is not enough user data to make accurate recommendations for new users, and the problem of data sparsity, which occurs when there is not enough data on certain books or users to make accurate recommendations. Additionally, privacy concerns related to user data collection and usage are an important issue that must be addressed when developing book recommendation systems.

This brief overview of book recommendation systems sets the stage for the subsequent sections of the literature review, which will dive deeper into the different types of recommendation systems and the challenges associated with developing effective deep learning models for personalized book recommendations.

### 2.2 Personalized Recommendation system

The emergence of personalized recommendation system can be traced back to the extensive research in cognitive science, approximation theory, information retrieval, forecasting theories, and also to management science related and to consumer choice modeling in marketing i.e., E-commerce. The recommender systems emerged as an autonomous research domain in the mid-1990s. This is the period where researchers started to focus on the recommendation problems that rely solely on ratings structure. The recommendation problem is reduced to the problem of recommender system to be estimating ratings for the items that have not been seen or rated by the user. Intuitively, this prediction or estimation is usually based on the user’s previous rated items. In this sense, if the estimate ratings for the yet unrated items can be obtain, then the item(s) with the highest estimated rating(s) can be predicted to the user (Gediminas, Alexander et al., 2005).

Personalized recommendation system is a computer-based system that offers individualized recommendations to users based on their previous behavior or history (Khatwani & Chandak, 2016). According to Dina F. et al. (2018), a recommender system in the context of education provides intelligent recommendations to students about study materials, courses, and related information based on their prior decisions. To provide such recommendations, the system employs a statistical model that is trained using a learning algorithm based on input from previous learning participants.

In a personalized recommendation system, users and items are the two main objects that play a crucial role. Users provide their interests about items, and this information is accumulated as input data in a utility matrix. The utility matrix represents the order of preference for particular items by the user in the combination of customer-item value. There are two classifications of recommender systems: user-based and item-based. In a user-based system, the user's interests, dislikes, and ratings are used to recommend unrated items. In contrast, an item-based system utilizes the relationship among items to generate recommendations for users (Geetha & Karthika, 2019).

Personalized recommendation systems play a vital role in both academia and industries. Many companies use personalized RS in their sales promotion, utilizing various platforms. For instance, the majority of the most-watched movies on YouTube and other online video databases come from the RS (Aminu & Naomie, 2019). The main purpose of building a recommender system is to provide maximum information required for the sole aim of personalizing learning and interests depending on the interactive patterns of users (Bouihi & Bahaj, 2019; Q. Zhang & D. Yang, P, 2020).

The algorithm for estimating the not-yet-rated items in the personalized recommender system is classified into three, namely: collaborative filtering, content base and hybrid approach (Kunal, Akshaykumar et al., 2017).

All authors agree that a personalized recommendation system is a system created with the major aim of recommending useful individualized information or items to users based on past learned historical records relating to the individual’s tastes or preferences.

### 2.2.1 Type of recommendation systems

There are three different types of traditional recommendation systems:

1. Collaborative filtering
2. Content base filtering
3. Hybrid approach

#### 2.2.1.1 Collaborative Filtering (CF)

The term collaborative filtering was coined in 1992 by Goldberg et al., who proposed that information filtering becomes more effective when human group relational preferences or involvements are consider while recommending items to users (Kunal, Akshaykumar, et al., 2017). This method of recommendation tries to predict the utility of items for a particular user based on the items previously rated by other similar users (Gediminas, Alexanda, et al., 2005). According to Kunal et al. (2017), collaborative filtering is a method of recommendation system in which recommendations are generated to various users by actively comparing the preferences of one active user with another who have qualified products similarly with the user in the past. Conversely, Sana et al. (2019) describes collaborative filtering as a technique that help make a recommendation by finding users similar interests to predict relatively. CF are based on the idea that people who agree with the evaluation of items in the past are likely to agree again in future. Collaborative filtering method are grouped into two general classes: neighborhood based and model-based (Gediminas, Alexanda, et al., 2005; Kunal, Akshaykumar, et al., 2017).

*A. Neighborhood Based Method*

The neighborhood-based method of collaborative filtering, also known as memory-based or heuristic-based CF, leverages the user-item ratings stored in memory to make predictions about the preferences for new items. This can be accomplished through two approaches: user-based and item-based recommendation (Qian, Jie, et al., 2020).

In the user-based approach, the interest of a user (referred to as "u") in an item (referred to as "i") is determined by considering the ratings given to that item by other users who exhibit similar rating patterns. These users are known as "neighbors". By examining the ratings of these neighbors, the system can estimate the likelihood of user u being interested in item i (Dhanashri, Nandani, et al., 2020).

On the other hand, the item-based approach involves predicting the rating that user u would assign to item i based on the ratings that user u has given to items similar to i. In this approach, the similarity between two items is calculated by examining the ratings provided by other users of the system. By analyzing these ratings, the system can identify items that are similar in terms of user preference and use this information to make predictions about user u's rating for item i (Bushra, Imran, et al., 2019).

Overall, the neighborhood-based method of collaborative filtering utilizes the existing user-item ratings to determine recommendations for new items. The user-based approach focuses on finding users with similar rating patterns, while the item-based approach concentrates on identifying similar items based on user ratings. By employing these methods, the system can provide personalized recommendations that align with user preferences.

Kunal et al. (2017) classified the main advantages of neighborhood-based methods as follow:

**Simplicity**: Neighborhood-based methods are relatively straightforward to implement, making them accessible and easy to work with.

**Efficiency**: This method does not require expensive training phases that need to be regularly performed in large commercial applications. This saves computational resources and time.

**Justifiability**: Neighborhood-based methods provide concise yet comprehensive explanations for their predictions. This makes it easier for users to understand why certain recommendations are made.

**Stability**: Despite the continuous addition of items, users, and ratings typically encountered in major e-commerce applications, neighborhood-based methods remain relatively unaffected. They maintain stability in their recommendations even as the system evolves over time.

*B. Model-Based Recommendation Methods*

The model-based recommendation method differs from neighborhood-based systems in that it doesn't directly use stored ratings for prediction. Instead, it utilizes these ratings to acquire knowledge and build a predictive model. In model-based approaches, the aim is to replicate and model the interactions between users and items, incorporating factors that represent the hidden characteristics of both users and items within the system. These factors can include user preferences and item categories, among others.

To implement a model-based system, the available data (usually in the form of a training dataset) is used to train the predictive model. The model is trained to understand the patterns and relationships between users, items, and their hidden characteristics. Once the model is trained, it can be applied to predict ratings for new items based on user behavior and item attributes.

Numerous model-based approaches exist for the task of item recommendation, each employing specific techniques. These techniques include Bayesian Clustering, which utilizes probabilistic clustering algorithms; Latent Semantic Analysis, which applies matrix factorization techniques to uncover latent factors in user-item interactions; Support Vector Machines, which employ machine learning algorithms for classification and regression tasks; and Singular Value Decomposition, which decomposes the user-item rating matrix into lower-dimensional representations (Nilashi, Bagherifard, et al., 2012; Geetha & Renuka, 2019)

These model-based approaches offer a more sophisticated and comprehensive way to make recommendations by capturing underlying patterns and relationships within the data. They leverage machine learning and statistical techniques to create predictive models that can effectively estimate user ratings for new items, enhancing the recommendation process.

#### 2.2.1.2 Content-based filtering (CBF)

Content-based recommender systems are widely used in information or book retrieval. The process involves manually assigning terms and selecting appropriate techniques to compare these terms with the information in the user's profile. A learning algorithm is then applied to perform the chosen techniques and provide relevant recommendations to the user (Javed & Shaukat, 2021).

Term frequency (TF) and inverse document frequency (IDF) are important concepts used in information retrieval and content-based filtering systems, such as content-based recommenders. These concepts determine the relative importance of a document, article, news item, film, etc. Content-based filtering is a common approach in recommender system design, relying on item descriptions and user profiles. In a content-based recommender system, keywords are used to describe items, and a personal profile is constructed to suggest items that align with the user's preferences. In other words, these algorithms aim to recommend items similar to those that the user has liked or currently shows interest in. To achieve this, multiple candidate items are compared with items previously rated by the user, and the best matching items are recommended. Yannick et al. (2010) employed a content-based algorithm to filter and recommend the best results to users. The advantages of content-based recommender systems include transparency, independence, and recommendations for unclassified entities. However, they also have drawbacks such as limited serendipity, partial content analysis, and overspecialization (Adomavicius, Tuzhilin, et al., 2005).

By leveraging a content-based recommender system, recommendations can be generated based on the user's provided data or implicit interactions with the system. As the user provides more data or accepts suggested activities, the system becomes increasingly accurate in generating personalized recommendations. Compared to collaborative filtering systems, content-based approaches offer easier implementation and have been used in many projects due to their ability to generate highly relevant and transparent results. Content-based filtering strategies often employ techniques for content extraction, characterization, and building user and item profiles. However, these techniques have limitations, such as mismatches between user profile items and profile terms, leading to lower performance. Notable examples of content-based recommendations include Fab, which suggests web pages, and ELFI, which recommends financial information from a database (Lops, Gemmis & Semeraro, 2011).

Content-based filtering systems adapt their behavior to individual users by learning their preferences from previously relevant documents. The process involves comparing representations of item content with representations of user interests to find the most relevant items for each user. This task requires finding the best representation for both items (item profile) and users (user profile). A user profile represents a mapping of the user's real-world interests to a simplified domain model, while an item's profile describes its content using keywords or other means of representation. By enabling matching between user and item profiles, a common representation is established (Javed & Shaukat, 2021).

Content-based recommendation systems operate by analyzing a collection of documents and/or descriptions of items that have been previously rated by a user. These systems construct a model or profile of the user's interests based on the features of the rated objects (Lops, Gemmis & Semeraro, 2011). As described by Mladenic (2018), the profile serves as a structured representation of user interests and is utilized to recommend new and interesting items. The recommendation process primarily involves comparing the attributes of the user profile with the attributes of a content object (Khatwani & Chandak, 2016). This comparison results in a relevance judgment that indicates the user's level of interest in the object. A well-constructed profile that accurately reflects user preferences greatly enhances the effectiveness of an information retrieval process (Geetha & Renuka, 2019).

An example of the profile's utility is its application in filtering search results. By leveraging the user profile, it becomes possible to determine whether a user would be interested in a specific web page. If not, the system can prevent the display of that particular page (Lops, Gemmis & Semeraro, 2011).

##### High-Level Architecture of Content-based Systems

Content-based Information Filtering (IF) systems require effective techniques for representing items and generating user profiles, as well as strategies for comparing the user profile with item representations (Adomavicius, Tuzhilin, et al., 2005). In their study, Lops et al. (2011) outlined a three-step recommendation process, with each step handled by a distinct component:

1. **Content Analyzer**: The Content Analyzer is responsible for processing information without a predefined structure, such as text, and extracting relevant structured information. Its main task is to represent the content of items (e.g., documents, web pages, news, product descriptions) from various sources in a format suitable for further processing. Through feature extraction techniques, the component analyzes data items and transforms their representation from the original information space to the target space, such as representing web pages as keyword vectors. The output of the Content Analyzer serves as input for the Profile Learner and Filtering Component.
2. **Profile Learner**: The Profile Learner module collects representative data on user preferences and generalizes this data to construct the user profile. Typically, machine learning techniques are employed to infer a model of user interests based on items previously liked or disliked. For example, in a web page recommender system, the Profile Learner can utilize a relevance feedback method (Rocchio, J., 1971), where vectors of positive and negative examples are combined to create a prototype vector representing the user profile. Training examples consist of web pages that received positive or negative feedback from the user.
3. **Filtering Component**: The Filtering Component leverages the user profile to suggest relevant items by comparing the profile representation with the representations of items to be recommended. This comparison results in a binary or continuous relevance judgment, often computed using similarity metrics. In the case of continuous relevance, a ranked list of potentially interesting items is generated. In the previous example, the matching process involves computing the cosine similarity between the prototype vector and the item vectors.

The recommendation process begins with the Content Analyzer, which applies techniques from Information Retrieval systems to process item descriptions from information sources. The Content Analyzer extracts features such as keywords, n-grams, and concepts from unstructured text to generate a structured representation of items stored in the Represented Items repository.

To construct and update the user profile for the active user (Ua) who requires recommendations, the system collects the user's reactions to items and records them in the Feedback repository. These reactions, known as annotations or feedback, along with the corresponding item descriptions, are used during the process of learning a model that predicts the relevance of newly presented items. Users can either provide explicit feedback, where they actively evaluate items, or implicit feedback, which is derived from monitoring and analyzing user activities (Lops, Gemmis & Semeraro, 2011).

Explicit feedback can be obtained through approaches such as like/dislike ratings, numerical ratings, or text comments. Implicit feedback methods assign a relevance score based on user actions, such as saving, discarding, printing, sounds, manipulation of physical objects, or bookmarking of items. While explicit feedback is straightforward, the adoption of numeric or symbolic scales can increase cognitive load. Implicit feedback methods do not require direct user involvement but may be subject to biases, such as interruptions during reading (Meddeb, Maraoui & Zrigui, 2021).

The Profile Learner uses a training set (TRa) for the active user (Ua), consisting of pairs ⟨Ik, Rk⟩, where Rk represents the rating provided by Ua for the item representation Ik. By applying supervised learning algorithms to the labeled item representations, the Profile Learner generates a predictive model, i.e., the user profile, which is stored in a profile repository for future use by the Filtering Component. When presented with a new item representation, the Filtering Component compares its features with those in the user profile to predict the user's interest. Typically, the Filtering Component employs strategies to rank potentially interesting items based on their relevance to the user profile. The top-ranked items form a recommendation list (La) presented to the user (Ua). As user preferences change over time, the user profile must be continually updated by incorporating up-to-date information. User feedback on the recommendations helps refine the learning process, creating a feedback-learning cycle that accommodates the dynamic nature of user preferences (Lops, Gemmis, & Semeraro, 2011).

In summary, Content-based Information Filtering systems involve the Content Analyzer for item representation, the Profile Learner for constructing user profiles, and the Filtering Component for suggesting relevant items based on user profiles. These components interact in a recommendation process that incorporates user feedback to adapt to evolving user preferences.

##### 2.2.1.3 Hybrid approach

### 2.3 Deep Learning Model