République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

Master Thesis in Computer Science

Presented by

BOUFAIDA SOUNDES OUMAIMA

Specialty: Intelligent Computer Systems

Theme

A HYBRID SYSTEM FOR RECOMMENDING BOOKS

Defence on 04 / 07 / 2021

Member of the jury:

Quality	First and second Name	Grade	University
President	Mr. BENMACHICHE A	MCA	Chadli Bendjedid El-Tarf
supervisor	**Mme. ANGUEL F**	**MCB**	**Chadli Bendjedid El-Tarf**
Examiner	Mme. MATALLAH M	MCA	Chadli Bendjedid El-Tarf

University Year: 2020/2021

Acknowledgment

This work is the culmination of hard work and a lot of sacrifice,

Our thanks go first to the Creator of the universe, who has endowed us with intelligence,

Courage and kept us healthy to carry out this year Study.

I also want to express my thanks to my family, and more specifically

My mother NADIA,

My father YAZID,

And my sisters HADJER, AYA, ISRA

To my grandparents and all my families.

Who have always supported me and pushed me to continue my studies.

This present work was made possible thanks to their support.

I express my gratitude to my memory director, Ms. ANGUEL FOUZIA,

Doctor at the University Chadli Bendjedid elTaref.

Thank you for his unwavering availability, his help as well as his judicious remarks,

Which allowed me to advance as well as enrich my work.

I express my deep appreciation and warm thanks to

The members of the jury who did me the honor of kindly evaluating and judging my work.

I want to share with all my family, all my dear friends

And all my colleagues. Of this memory

Dedication

I dedicate this work:

To my father

For his great love, his patience, his encouragement, his sense of duty and

His sacrifices so that I succeed in my studies.

To my mother

For his affection, his patience, his encouragement during difficult times as well as his

Prayers, which bring me happiness and success.

My sisters

For their support and encouragement.

To my grandparents and all my families.

To all of those who love me and to all of those I love.

To everyone I know from near or far.

Contents

List of Figures

List of paintings

List of acronyms

RS	Recommander system
CF	Collaborative Filtering
TF-IDF	Term Frequency-Inverse Document Frequency
CBF	Content-based filtering
DFS	Demographic Filtering Systems
LDA	Dirichlet latent allocation

Recommendations typically speed up searches thereby making it easier for users to access content that they are interested in, and surprise them with offers they would have never searched for. Recommender systems in recent years have become extremely common and are applied in a variety of popular applications. The most famous ones are probably movies, music, news, books and products in general. Over the years, collaborative filtering had emerged as the most prominent approach for recommendations. There has been an explosion of methods that are introduced in the area of recommendations, in the recent years. Moreover, the development of recommender systems has also increased the complexity of the modern systems when compared to the traditional or basic systems that utilize methods such as collaboration and content based filtering.

In the era of information overload, Internet users may find it difficult to choose from the multitude of available products and services. There is a need for recommender systems (RSs) that make personalized suggestions. The idea behind RSs is not new. It is common to ask acquaintances for recommendations when one chooses a restaurant, movie, book and so on. An RS predicts how likely the target user is to be interested in an item possibly not known to her yet. In order to make a recommendation, an RS usually needs user data, items, and user feedback on those items. After making a suggestion, user feedback on the item is acquired either explicitly or implicitly. The system stores the feedback in a database, and uses it for future recommendations [Ricci, F et al 2011].

Many studies have shown the benefits of reading. In a comparison between the wellbeing of 7500 Canadian adult readers versus non-readers, the former have been found statically significantly more likely to report better health/mental health, to volunteer and to feel strongly satisfied with life [Hill, K. 2013]. Such statistics support the statement by the National Institute of Child Health and Human Development (NICHD): “Reading is the single most important skill necessary for a happy, productive, and successful life.”

Fiction has also been found to stimulate profound social communication [Mar, R.A et al ,2008]. Exposure to fiction correlates with a greater ability for empathy and social support [Mar, R.A et al ,2009].It also has a biological lingering effect, especially in the connectivity of the brain [Berns, G.S et al, 2013] .Consequently, it is a worthwhile endeavour to deploy the techniques of artificial intelligence in order to spark people’s interest in books by recommending the right type of books. People trust RSs; one study [Chen, Y.-F. 2008] shows that consumers were more

interested in books labeled “customers who bought this book also bought” than books marked “recommended by the bookstore staff”.

. Thus, In this work we propose a recommendation system, which is based on topic modeling, ranking and graphs for book suggestion. The proposed model uses the textual elements of the books (description, titles, author book_id) to represent books ,as well as user profile with a set of topics based on the use of LDA (Latent Dirichlet Allocation), also we have integrated the raking method to improve our recommendation system and finally the recommendation are displayed as a graph to focus on the more relevant books.

This dissertation is organized into four chapters.

The first chapter (Background of recommender systems):

Is devoted to the presentation of basic concepts on recommendation systems and the classification of existing recommendation approaches . We mainly expose the methods of recommendation, their problems as well as the solutions and the evaluation of recommender systems..

The second chapter (Book recommender systems):

Is a review of some works for book recommender systems specially, where for each work the used data and technique of recommendation is explained.

The third chapter (Design of the proposed approach):

Is dedicated to a detailed conceptual study of the proposed system by presenting the main objective of our work, the architecture of our work and the different stages of the used.approaches

The last chapter (Implementation and exprementation):

Presents the tools and languages used for the development. Of our recommendation system as well as steps of implementaio are detailed..

Finally, we end our dissertation with a general conclusion and some perspectives.

Chapter 1
State of the art

1. Introduction

Users no longer have the time to review all of the information available to them due to the rising number of products and services available on the Internet. Instead of allowing the user to waste time searching for the information he requires, it has become critical to build systems that make his work simpler by continuously offering content that piques his interest [Burke, 2002]. Recommendation systems then appeared as a specific field of research since the 1990s.

This chapter is an overview of recommender systems. This overview presents the definition of recommendation systems, followed by the presentation of several classification logics of these systems, well known in this field. Next, we outline the book recommendation approaches. Finally, of this chapter we present the different method work of recommendation systems.

2. Definition of Recommender Systems

Recommender Systems are software tools and techniques of machine learning that provides suggestions for items to an individual user. Recommender systems enable an improved access to relevant products and information by making personalized suggestions based on the examples of a similar user's likes and dislikes. Recommendation systems emerge into intelligent algorithms, which can generate results in the form of recommendations to the users. The popular suggestions are related to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. Whatever the system suggests the user, is termed as, the "item". A system normally focuses on a specific type of item, say product or utility range and accordingly its design, its graphical user interface, and the core recommendation technique used to provide the recommendations are all trained to generate useful and effective suggestions for that unique type of item. Recommendation systems are initially directed towards individuals who lack sufficient personal experience to evaluate the overwhelming number of alternative items that a resource might offer.

3. Basic Concepts, Notation, and Related Notions

In this section, we define some concepts related to recommender systems:

3.1. User and Items Entities

In any recommendation system, there are two important entities which are users and items.

The user is a person who accesses the system is registered, entering their demographic information, interests and other personal information. The set of users in the system is represented by *U*, where a given user is $u \in U$ [Maatallah, 2016].

An item is the entity which represents any element constituting a recommendation list and which corresponds to the user's needs, including any product likely to be sold (book, products, ... ect in the e-commerce sites such as Amazon.com), viewed (movies in online TV sites such as Netflix), listened to (music), or read (such as news in online newspapers, magazines in digital libraries), as well as vacation destinations, restaurants, etc. An item can also be an individual or a set of individuals suggested to the user in social networks. The set of items available in the system is represented by *I*, where $i \in I$ [Maatallah , 2016].

3.2. Evaluation (Score or Vote)

An evaluation is a numerical value in any scale (the most used is [1-5]) or binary (like \ Dislike, good \ bad, etc.) which represents the preference or not of an item given by a user. The evaluation given by a user u to an item i is represented by a triplet <u, i, r>. Where, a rating of 5, for example, expresses a high preference and a rating of 1 indicates a low preference i.e. the user didn't like the item.

A rating can be assigned directly by a user to an item by giving a numerical or binary value through the system interface called an explicit rating. In addition, user preferences can be inferred by the system using specific algorithms and techniques, and in this case called implicit evaluation [Burke ,2002].

3.3. User-Item Rating Matrix

The set of all the triples of the system <u, i, r> are recorded in a sparse database called the rating matrix or even the user-item Matrix and it is noted by *R*, where each row refers to a single user's ratings, and each column relates to all users' ratings for a particular item.

Several researches have arisen that propose novel possibilities for bringing new information to the rating matrix in order to improve the performance of recommender systems [SHI et HAN, 2014]. Depending on the source of the data or its relationship to system interaction, it can be split into two groups. The first one is rich literal information about users (gender, age, hobbies,..etc.) or on items (category, content,..etc.). The second type of information regroups the time of evaluation or purchase of items, the location (local) of the user as well as user comments and opinions [LEE et al., 2014].

3.4. Concept of community

A community or group is a collection of users with similar likes and tastes who are put together according to a common characteristic. The community building process can use a variety of factors, such as item's content reviewed by the users, their ratings of the items, their demographic information, and their area of interest [BOU, 2005]. The system's communities vary according to each of these parameters. . As a result, any user can be a member of as many communities as the training criteria allow [NGU et al., 2006].

3.5. Notion of profile

- User Profile

User profile is a description of the user's attributes, which could include their areas of interest, demographics, or preferences represented as ratings, among other things. There are several methods for gathering information about a user in order to create a profile, which can be divided into manual and automatic or semi-automatic approaches [Burke, 2002].

- Item profile

In a book recommendation system, for example, the items (books) are represented by their Ids, title, genre, Author, Editor, year of publication, which forms a user profile. Therefore an item profile is the description of items with a set of properties, also called attributes or characteristics. Characteristics vary according the item's nature, keywords can describe the semantic content of a document [RIJ, 1979].

3.6. Prediction

Prediction is the estimation of the user's likely rating for an item that he has not yet viewed or rated. The prediction is calculated using the scores provided by the user's neighbors (user-based prediction) or assigned to the neighboring items (item-based prediction), or given by a model (model-based prediction). Then the items with the highest prediction values will be recommended to the user.

3.7. Recommendation

The recommendation is the act of calculating a list of items (Top-N items) that the user will like the most. The calculation of the recommendation lists is done by assigning scores for the items according to their popularity or their preferences [WEN et al. 2008]**.** Nethereless, unlike prediction, the calculation of recommendations is not based strictly on ratings.

3.8. Customization

Customization or personalization consists in adapting an item to the user's preferences, needs and sometimes even to the behavior of the user. While a recommendation generates a list

of items more or less suited to the user's needs. Thus, personalization is similar to recommendation, but it is less general because recommendation, generates a list of items that are more or less appropriate to the user's demands.

4. Types of Recommender Systems

Recommender Systems are primarily categorized on the basis of personalized recommendations and non-personalized recommendations. Personalized recommendations are offered as ranked list of items. Personalized recommender systems are used by E-commerce sites to recommend products to their customers (based on past activities of the individual). Non-Personalized recommendations which are must simpler to generate and are normally featured in magazines or newspapers. Non–personalized recommender system recommends products to customer based on what other customers have said about the products on average. The recommendations are not dependent on the users, so each customer gets the same recommendation.

Recommender Systems can be classified broadly into several categories depending on the information they use to recommend items:

- Content-based Filtering Systems: Uses information of active users and data about the items.
- Collaborative Filtering Systems: Uses information about a set of users and their relations with the items to provide recommendations to the active user.
- Demographic Filtering Systems: Uses demographic information such as age, gender, education, etc. of people for identifying types of user.
- Hybrid recommender Systems: By putting as a forward feature, it uses combination of Content-based and Collaborative filtering.

4.1. Content-based Filtering

Content-based filtering(CBF) recommender systems recommend items based on contents of items rather than other users rating of the system. Instead of using a user-to-item correlation and defining methodologies, they use item-to-item correlation for generating recommendations. Following steps are carried out in the process of generating recommendations:

- ✓ Gathering content data about the item (For example- title, author, cost etc. for the books are some of the common content information.
- ✓ Process data and extract useful features and elements about its content.

A. Advantages

Content based approach doesn't require data of other users and has capabilities of recommending items to user with unique taste. It avoids first rater problem.

B. Disadvantages

In content based filtering items are limited to their initial descriptions or

4.2. Collaborative Filtering

Collaborative Filtering (CF) assumes that if two users have similar rating history, their future ratings will be similar as well. This method uses the available ratings of active users to predict the preferences of other users. CF comes in two forms. User-based CF finds the similarity between users. Item-based CF computes the similarity between two co-rated items (rated by common users) [Sarwar et al., 2001]. To make recommendations, CF only requires an item-user rating matrix as depicted in figure(1).

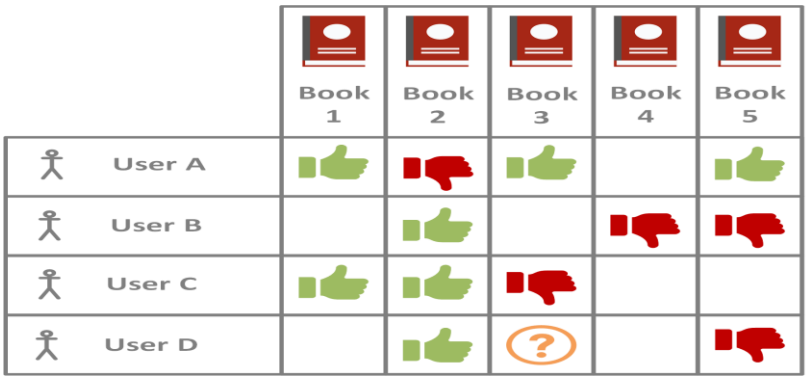


Figure 1: User-Item matrix for Collaborative Filtering

Techniques adopted in CF are categorized into memory-based and model-based. Neighborhood based CF, which falls under the former category, calculates the similarity between two users or two items and predicts ratings by computing the weighted aggregate of nearest neighbors' ratings. There are many similarity measures including Pearson correlation and cosine similarity. From the set of nearest neighbors, the CF recommends the most relevant predicted items as a ranked list [Su et al, 2009].

One common model-based algorithm is matrix factorization (MF) that represents users and items in a space where each item/user is modeled as a vector of latent factors. A user-item interaction is characterized as an inner product in the space. The predicted rating is the dot product between the user and item vectors [Koren et al., 2009]. Matrix factorization, which we consider as a baseline, provided more accurate top-k recommendations when compared with neighborhood algorithm in [Hu et al.,2008].

In Collaborative filtering, recommendations are based on a few users who are most similar to the active users. It computes the similarity of two users in various ways; one common method is to calculate the cosine of the angle between the two vectors [Sarwarm et al., 2012]

The collaborative filtering can be adapted with neighbourhood methods, whose focus is on relationship between the items or, alternatively between the users. They are:

4.2.1. User-based CF

For each user, compute correlation with other users. For each item, aggregate the rating of the users highly correlated with each user. Problem: Sparsity, easy to attack

4.2.2. Item-based CF

For each item, compute correlation with other items. For each user, aggregate his rating of the items highly correlated with each item

A. Advantages

Collaboration filtering approach doesn't need a representation of items in terms of features but it is based only on the judgment of participating user community.

B. Disadvantages

The item can't be recommended to any user until and unless the item is either rated by another user(s) or correlated with other similar items.

4.3. Demographic Filtering Systems

It uses pre-existing knowledge of demographic information about the users and their opinions for the recommended items as a basis for recommendations. Demographic systems (DFS)are stereotypical, because they depend on the assumption that all users belonging to a certain demographic group have a like taste or preference.

A. Advantages

It does not require history of user ratings that are required by collaborative and content-based techniques. This is a quick, easy and straight forward approach for making results based on few observations.

B. Disadvantages

Concerning the security and privacy issue, gathering of complete user information is impractical.

4.4. Hybrid Recommender Systems

It is another category of recommender system that tries to overcome the limitations of the other approaches discussed before. It is a combination of two or more different recommendation techniques. The most popular hybrid approaches are those of content based and collaborative filtering. It use both item content and the ratings of all users [Prasad et al.,2012]

4.5. Context-Aware Recommender System

It is one of the most trending recommender systems these days. It helps in giving diverse and accurate recommendations to the user. The contextual information may include location of the user, Identity of people around, date, season, temperature etc.[Adomavicius, 2011] The contextual information may be retrieved in a number of ways, including:

- Explicitly : gathering information by asking the direct questions from the user. For example, a website may recommend songs to a user by asking the current mood of the user.

- Implicitly : from the data or the environment,

- Inferring (To conclude from evidence or by reasoning).

5. Challenges and Solutions

The web recommender system suffers from many challenges such as lack of data, changing data, changing user preferences, unpredictable items, scalability, privacy protection some of them are: cold-start, privacy, over-specialization, scalability,sparsity, freshness etc.

5.1. Challenges of recommender systems

5.1.1 Cold-start

Cold start problem can be classified into two categories, cold start of new items and cold start of new users. Cold start problem for an item occur when we don't have enough previous rating related to that item. Also, it is a bit difficult to recommend items to new users as the system don't have any information related to his past purchases or it might be possible that he has not rated any item yet so his taste is unknown to the system.

5.1.2 Scalability

As the numbers of users and items grows the system needs more resources in order to give the most accurate recommendations to the users. Most of resources are used in the purpose of determining users of similar tastes, and items with similar attributes. It is one of the problems found in collaborative filtering approach.

5.1.3 Sparsity

Sparsity is the problem of lack of knowledge. Suppose, for an online shop that has a huge amount of users and items. If a user purchased few items from the shop and has rated any of them. Then, it will lead to the problem of sparsity.

5.1.4 Privacy

Privacy is also a big issue in context of demographic recommender systems. In order to give the most accurate recommendation to the user, the system must acquire the most appropriate information of user, including demographic data (age, sex, email-id, hobbies etc.), and data about the location of a particular user which may breach the privacy of the user.

5.1.5 Over-Specialization

This is one of the most common problems faced by the content-based recommendation system. A good recommender system must suggest diverse items which content-based system lacks. It gives nothing "surprised". It hinders the users from discovering something new and different. Users are recommended items they are already familiar with.

5.1.6 Freshness/Predictability

One more problem generally faced by recommender systems these days is that of predictability. Even if the items recommended to the user are diverse, it might be familiar to the user. For example, a system recommends best sellers only. The recommendation in this case are indeed diverse but the user may already know or be familiar with the recommended stuff.

5.2. Possible Solutions

5.2.1 Cold-Start

To curb the problem of cold-start, we can use the demographic information of the user from social networking sites or through the sign up page of the website. Also, we can use hybrid approach, i.e. to use collaborative filtering with demographic recommending approach to suggest items to a new user.

5.2.2 Sparsity

The problem of Sparsity can be resolved using hybrid recommendation technique. Instead of using content based alone we can combine the content based and collaborative technique together which will result as a solution of Sparsity. The amount of information people have in common can be increased by using the attributes of an item instead of the item itself [].

5.2.3 Overspecialization

The problem of overspecialization can be overcome with the neighborhood based

collaborative filtering technique. For the probabilistic neighborhood selection phase, we use a method for weighted sampling of k neighbors that takes into consideration the similarity levels between the user/items and candidate neighbours [Panagiotis et al, 2014].

6. Phases of Recommendation Process

The recommandation process is composed of three phases , namely :

- ✓ Information collection phase
- ✓ Learning Phase
- ✓ Prediction /Recommandation phase

Figure 1 illustrates the recommendation phases.

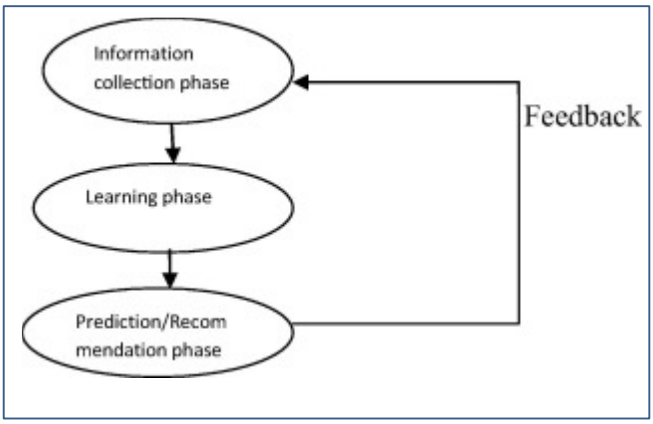


Figure 2. Recommendation Process [Shubham Gupta et al , 2020]

6.1. Information Collection Phase

This collects relevant information of users to generate a user profile or model for the prediction tasks including user's attribute, behaviors or content of the resources the user accesses. The system needs to know as much as possible from the user in order to provide reasonable recommendation right from the onset. Recommender systems rely on different types of input such as the most convenient high quality explicit feedback, which includes explicit input by users regarding their interest in item or implicit feedback by inferring user preferences indirectly through observing user behavior [Oard, 1998]. Hybrid feedback can also be obtained through the combination of both explicit and implicit feedback. The user profile is normally used to retrieve the needed information to build up a model of the user. Thus, a user profile describes a simple user model. The success of any recommendation system depends largely on its ability to represent user's current interests. Accurate models are indispensable for obtaining relevant and accurate recommendations from any prediction techniques.

A. Explicit feedback

The system normally prompts the user through the system interface to provide ratings for items in order to construct and improve his model. The accuracy of recommendation depends on the quantity of ratings provided by the user. The only shortcoming of this method is, it requires effort from the users and also, users are not always ready to supply enough information. Despite the fact that explicit feedback requires more effort from user, it is still seen as providing more reliable data, since it does not involve extracting preferences from actions, and it also provides transparency into the recommendation process that results in a slightly higher perceived recommendation quality and more confidence in the recommendations [Buder et al., .2012].

B. Implicit feedback

The system automatically infers the user's preferences by monitoring the different actions of users such as the history of purchases, navigation history, and time spent on some web pages, links followed by the user, content of e-mail and button clicks among others. Implicit feedback reduces the burden on users by inferring their user's preferences from their behavior with the system. The method though does not require effort from the user, but it is less accurate. Also, it has also been argued that implicit preference data might in actuality be more objective, as there is no bias arising from users responding in a socially desirable way [Buder et al., .2012].

C. Hybrid feedback

The strengths of both implicit and explicit feedback can be combined in a hybrid system in order to minimize their weaknesses and get a best performing system. This can be achieved by using an implicit data as a check on explicit rating or allowing user to give explicit feedback only when he chooses to express explicit interest.

6.2. Learning phase

It applies a learning algorithm to filter and exploit the user's features from the feedback gathered in information collection phase.

6.3. Prediction/recommendation phase

It recommends or predicts what kind of items the user may prefer. This can be made either directly based on the dataset collected in information collection phase which could be memory based or model based or through the system's observed activities of the user.

7. Application of recommender systems

Recommender systems application have been developed in various domains. For the most common RS applications we cite :

- Entertainment - recommendations for movies, music, and IPTV.
- Content - personalized newspapers, recommendation for documents, recommendations of Web pages, e-learning applications, and e-mail filters.
- E-commerce - recommendations for consumers of products to buy such as books, cameras, PCs etc.
- Services - recommendations of travel services, recommendation of experts for consultation, recommendation of houses to rent, or matchmaking services.

8. Evaluation of Recommander System

In all experimental scenarios, it is important to follow a few basic guidelines in general experimental studies:

- ✓ Hypothesis: before running the experiment we must form an hypothesis. It is Important to be concise and restrictive about this hypothesis, and design an experiment that tests the hypothesis.
- ✓ Controlling variables: when comparing a few candidate algorithms on a certain hypothesis, it is important that all variables that are not tested will stay fixed.
- ✓ Generalization power: when drawing conclusions from experiments, we may desire that our conclusions generalize beyond the immediate context of the experiments.

The evaluation of recommender systems generally follows one of three methods: Offline evaluation, sample user studies or online evaluation.

8.1. Offline Experiments

An offline experiment is performed by using a pre-collected data set of users choosing or rating items. Using this dataset, we can try to simulate the behavior of users that interact with a recommendation system. In doing so, we assume that the user behavior when the data was collected will be similar enough to the user behavior when the recommender system is deployed, so that we can make reliable decisions based on the simulation. Offline experiments are attractive because they require no interaction with real users, and thus allow us to compare a wide range of candidate algorithms at a low cost.

8.2. User Studies

Many recommendation approaches rely on the interaction of users with the system. It is very difficult to create a reliable simulation of users interactions with the system, and thus, offline testing are difficult to conduct. In order to properly evaluate such systems, real user interactions with the system must be collected. Even when offline testing is possible, interactions with real users can still provide additional information about the system performance. In these cases we typically conduct user studies.

A user study is conducted by recruiting a set of test subjects, and asking them to perform several tasks requiring an interaction with the recommendation system.

8.3. Online Evaluation

In many realistic recommendation applications the designer of the system wishes to influence the behavior of users.We are therefore interested in measuring the change in user behavior when interacting with different recommendation systems. For example, if users of one system follow the recommendations more often, or if some utility gathered from users of one system exceeds utility gathered from users of the other system, then we can conclude that one system is superior to the other, all else being equal.

The real effect of the recommendation system depends on a variety of factors such as the user's intent (e.g. how specific their information needs are, how much novelty vs. how much risk they are seeking), the user's context (e.g. what items they are already familiar with, how much they trust the system), and the interface through which the recommendations are presented.

Thus, the experiment that provides the strongest evidence as to the true value of the system is an online evaluation, where the system is used by real users that perform real tasks. It is most trustworthy to compare a few systems online, obtaining a ranking of alternatives, rather than absolute numbers that are more difficult to interpret.

In any type of experiment it is important that we can be confidant that the candidate Recommender that we choose will also be a good choice for the yet unseen data the System will be faced with in the future.

9.1.Content-Based Filtering (CBF)

In book recommendation systems for students, the items are the books in the digital library and the users are the students. In the CBF method, first the student's books are collected. A student's books or other information is used to build their profile.

There are many ways to build a student's profile [Bai et al, 2019]. For example, a student's preferences and interests can be represented by extracting keywords from the student's area of research. Additionally, book recommendation systems can extract keywords from the title, abstract, and content of books to represent those books. Candidate books can be retrieved from the digital library. The RS then calculates the keyword similarity between the user profile and the candidate books and then sorts them. Finally, candidate books with a high similarity will be recommended to the student. According to the underlying logic. The CBF system extracts information from the books and compares them. If the book is related to the interests of the students, it will be discovered. Also, compared to keyword-based search engines, CBF generally considers the current interests of the book, and does not involve other students.

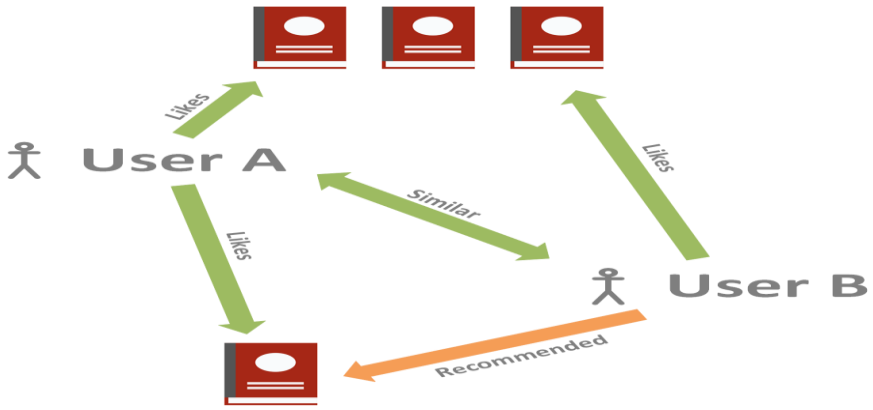


Figure3: Content-based filtering [TORI TOMPKINS, 2020]

Vaz et al. presented a content-based book recommendation prototype that searches for books similar to books written by a given author. The prototype represents books as feature vectors and uses content-word frequencies to retrieve books with similar subjects. To retrieve books with similar writing styles, the prototype uses stylometric features. Stylometric features present a promising ground in the book recommendation field, because they are subject independent and allow the retrieval of books that are more similar to the type of books the user likes to read. This is an important feature in literary book recommendation **[**Vaz et al., 2012b**]**

Alharthi proposed content-based recommender systems that extract elements learned from book texts to predict readers' future interests. They proposed a system that recommends books after learning their authors' writing style. Another approach that they proposed uses over a

hundred lexical, syntactic, stylometric, and fiction-based features that might play a role in generating high-quality book recommendations. Their content-based systems suffer from the new user problem, well-known in the field of RSs that hinders their ability to make accurate recommendations. Therefore, they proposed a Topic Model-Based book recommendation component (TMB) that addresses the issue by using the topics learned from a user's shared text on social media, to recognize their interests and map them to related books. Using topic modeling techniques, extracting user interests can be automatic and dynamic, without the need to search for predefined concepts. Though TMB is designed to complement other systems, we evaluated it against a traditional book CB. They assessed the top k recommendations made by TMB and CB and found that both retrieved a comparable number of books, even though CB relied on users' rating history, while TMB only required their social profiles. [Alharthi, 2019]

Monney et al. used content-based book recommending technique for text categorization. In the paper researcher explored the development of a widely-used recommendation system by using collaborative filtering which used others user's interests to recommend individual users. However, it was found that the recommendation done by using content-based filtering yielded better results because it was directly based on the product. [Monney et al, 2000]

9.2. Collaborative Filtering (CF)

Vaz et al. proposed an item-based CF, this system calculates the cosine and Euclidean distance in users-books and users-authors rating matrices. The author RS and the book RS were evaluated using the LitRec dataset. The prediction performance was the best when 10% of author RS and 90% of book RS were merged [Vaz et al. 2012a].

Vaz et al. assessed the temporal relevance of ratings in item-based CF. Experiments on a closed version of the LitRec dataset found that high prediction errors resulted from using only recent ratings but neglecting early one (In a closed dataset, the rating matrix has no missing values: every user rates every book.) .It was also found that for recommendations of good quality one needs all ratings of the community but only recent ratings of the target user. **[** Vaz et al. 2013**]**

Ritu Rani et al. use a variety of algorithms like k-mean clustering, collaborative filtering. The proposed work implies that to maintain quality and authenticity of any system, information and data source are required. Similarity for calculating distance between user and cluster center is adjusted and calculated. While calculating mean, the user who gave scores are only considered. K-mean clustering improves the accuracy of clustering algorithm and is suitable with collaborative filtering when compared. Rating scales for items are different in collaborative filtering for different user. Most of the people gives low scores many gives high. Similarity

calculations for the factors are not considered but can be adjustable to overcome the defects. The approach of using it can provide with the solution balancing the average score and adjust the similarity using method of K-mean clustering. User is assigned with the most similar cluster of his search, while computation of similarity in user and cluster [Ritu Rani et al, 2017]

Said, A et al fair comparative evaluation of three selected frameworks. Thus, to achieve this goal they benchmarked their comparison by controlling the evaluation dimensions, by using the same data set as explained and by selecting the same recommendation methods, algorithms and metrics. The common recommendation approach implemented by the RS frameworks is the collaborative filtering. They implement memory-based CF and model-based CF methods. A full list of the common methods within the frameworks is given in [Said, A et al, 2014].

9.3. Hybrid Method

Research shows that recommendations comprise a valuable service for users of a digital library. While most existing recommender systems rely either on a content-based approach or a collaborative approach to make recommendations, there is potential to improve recommendation quality by using a combination of both approaches (a hybrid approach). Hybrid recommenders can consist of a combination of any of the above-mentioned techniques and other machine learning techniques much like the Graph Recommender and Regression approach. If done correctly, you can combine the positives of the above techniques and reduce the negatives such as ‘cold start’ and speed. State of the art recommendation systems rely heavily on a combination of collaborative based filtering and content-based filtering.

Huang et al . a Hopfield net algorithm was used to exploit high-degree book-book, user-user and book-user associations. Sample hold-out testing and preliminary subject testing were conducted to evaluate the system, by which it was found that the system gained improvement with respect to both precision and recall by combining content-based and collaborative approaches. However, no significant improvement was observed by exploiting high-degree associations. [Huang *et al*, 2002].

Vinodhini et al. develop a recommendation engine which can recommend books to users with increased accuracy by analyzing user interest and characteristics of books. A hybrid recommendation system is developed which becomes its user input in the form of ratings. This list of ratings and user profile are the key terms used to predict user interest. The data set considered is a large set of books which is big data. In order to analyze the functionality of such a large set of books. The recommendation system that was developed has a special feature called Region Aggregation (RA). The user is prompted to enter country, state, and city details.Users are

grouped together using the K-means clustering algorithm. The profile of the users is considered to form the cluster. [Vinodhini et al, 2014]

Mercy Milcah Y et al. uses one of the filtering techniques known as collaborative filtering (CF) and content- based filtering, making the system a hybrid recommender system. The first method considered is the collaborative filtering under which, is a model base CF method called matrix factorization that is mainly used in this system to provide a personalized recommendation. Initially, the respective rating values are presented to the reader as the recommended set of books. Next is where, each of the books in the top 'N '(here, 5 is considered as 'n') book- list is known of its lexile score for reading where, the books with context similarity is listed. Accordingly, two or more books with similar context and lexile measurement are identified relatively. And this step does not require the ratings and reviews. Consequently, a list of efficient and closely relating books of favour is provided for each user.

This hybrid recommendation technology used has thus proved to be effective in suggestions, and useful in providing personalized recommendations. This system can be used in websites and book stores to be applied for user interaction and service providence. [Milcah et al,2020]

Rathnavel et al. try to present a model for a personalized recommendation system for books that uses hybrid recommendation approach which is combination of content based and collaborative filtering. The proposed recommendation system tries to learn the user's preferences and recommends the books to the user based on their preferences. The system also recommends the books to the user based on the user's demographic parameters like age and location. The system also tries to understand the user's favourite author and recommends accordingly. The proposed model tries to eliminate the problems like cold start problem by using demographic based recommendation, overspecialization problem by using light model which tries to predict books in such a way that the recommendation list contains book which has not been explored by the user yet. [Rathnavel et al, 2014]

10.Conclusion

In this chapter, we have presented recommender systems, the main recommendation techniques, their limitations and we have explained the related work of researchers in the book recommendation system and the different algorithms used.

We have also exposed the fields of application of recommendation systems as well as the different steps to evaluate a recommendation system.

Chapter 2
Design of the proposed System

1. Introduction

The realization of a system generally follows methods, which lead to model and build products reliably. It is with this objective that In this chapter, we discuss the architecture of our system Linked to the book recommendation system and the different steps to model our system using two search methods topic modeling using LDA with ranking and graphs.

2.1. Motivations

the recommendation systems aim to offer users items related to their current consultation and which may retain their interest. the interest of users depends on the context in which they find themselves. in this work, we propose a hybrid system that combines the context-sensitive recommendation system and graph-based recommendation. the context is here defined as the objective or the intention of the user.

2.2. Problématique

proposer un système de recommendation des livres à base des descriptions textuelle ainsi que les évaluations de ces livre

3. Proposed Approach

A recommendation system must make predictions that fit the interests of users. In this work, we combined the LDA method with graphs to improve the quality of the recommendations. These two methods were used because of their effectiveness and performance proven through the literature. The general architecture of the recommendation system in this project and the used method to realize this system are presented in the following sections.

3.1. Used Methods

In this section we present the different techniques that we have used for this work.

3.1.1. Content-based Filtering

The fundamental of the content-based approach is the features present in the user profile and the item description. For book recommender systems, the text data are dominant in in item's (book) description, namely, title, abstract, content,...etc. In the proposed approach we focus on exploiting this rich data to recommend relevant books. In the first step, our goal is to extract features for books representation. In fact, words can directly act as features. However, a better

representation of features in the text is topics. Topics are defined as the most significant words used in the text corpus. Using topics as features give an advantage in content based recommendation engines – some of the low occurring terms might seem irrelevant with respect to a book recommendation but these terms might be linked with other high-frequency terms, which are the strong features.

3.1.2. Topic Modeling with LDA

Briefly, Topic Models are statistical language models that are used to discover hidden structure in a collection of texts. In a more practical sense, Topic modeling is considered as task dimensionality reduction, an unsupervised learning task as well as :

- ✓ Dimensionality reduction where rather than representing a text T in its feature space as {Word_i: count (Word_i, T) for Word_i in Vocabulary}, The text is represented in a topic space as {Topic_i: Weight (Topic_i, T) for Topic_i in Topics}.
- ✓ Unsupervised learning, where it can be compared to clustering, as in the case of clustering, the number of topics, like the number of clusters, is an output parameter. By doing topic modeling, we build clusters of words rather than clusters of texts.
- ✓ Tagging, abstract "topics" that occur in a collection of documents that best represents the information in them.

There are several existing algorithms used to perform the topic modeling. The most common of it are Latent Semantic Analysis (LSA/LSI), Probabilistic Latent Semantic Analysis (pLSA), and Latent Dirichlet Allocation (LDA).

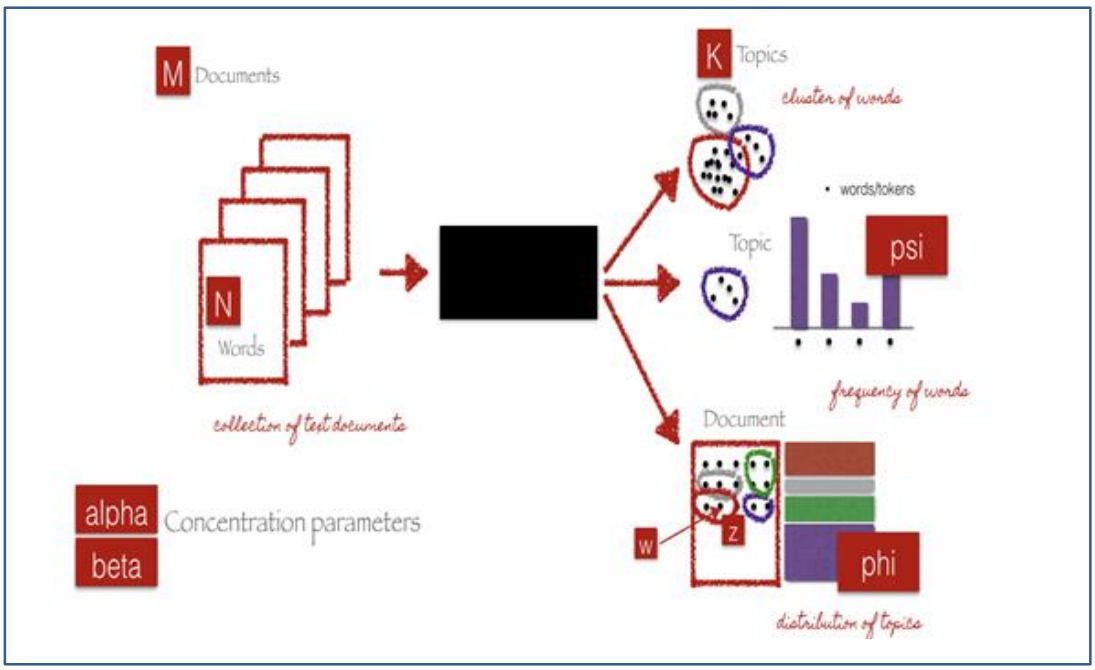


Figure4 Graphical representation of the LDA topic modeling [2019]

3.1.3. LDA algorithm

The Dirichlet latent allocation (LDA) model is a probabilistically generative theoretical model. It is based on the assumption that texts are made up of several topics (rather than words), with each theme being a multinomial distribution over a fixed vocabulary W.

The goal of LDA is to discover the topics that exist within a set of documents. The collection's documents are modeled as a collection of K topics, each of which is a multinomial distribution on W. The topic's distribution φ_k of a topic k is generated by a Dirichlet law with a parameter β, whereas the distribution θd of a document d is generated by Dirichlet law with a parameter α [Griffths, 2004]. A detailled description of this algorithm is as follows :

We denote by D the number of documents, K as the number of topics and N_d the number of word in document d. We define the following variables:

- $\varphi_{1:K}$: topic distributions over the vocabulary, where φk is the distribution for topic k.
- $\theta_{1:D}$: document distributions over topics, where θd is the distribution for document d.
- $z_{1:D,1:Nd}$: topic assignments for each document, where $z_{d,n}$ is the topic assignment for a word in position n of a document d.
- $w_{1:D,1:Nd}$: word occurrences for each document, where $w_{d,n}$ is the word that occurs in position n of document d.

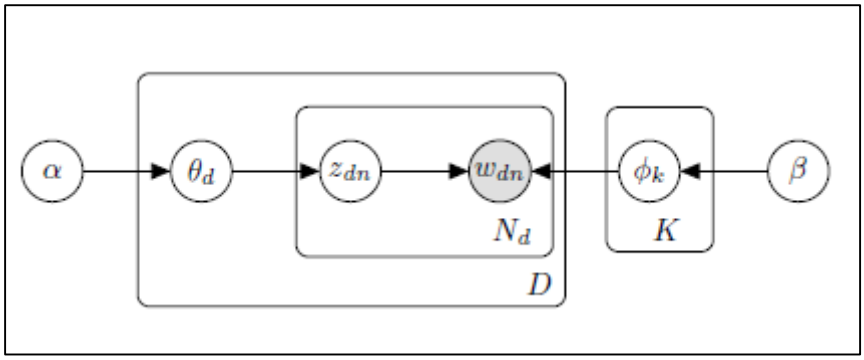


Figure Plate notation of the LDA model [Blei, 2012]

The generative process of a generic document d consists of the following steps:

- a topic distribution θ_d is randomly generated.
- for each word position in d
  - Randomly choose a topic k from θ_d.
  - Randomly choose a word w from φ_k.

The generative process of LDA corresponds to the following joint distribution of the hidden and the observed variables [Blei, 2012]:

$p(\phi_{1:K}, \Theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{k=1}^{K} P(\phi_K), \prod_{d=1}^{D} P(\Theta_D) \prod_{n=1}^{Nd} P(Z_{n,d}|\Theta_d) p(w_d, n|\phi_{1:K}, Z_{n,d})$
(formula1)

The joint distribution is used to compute the conditional probability of hidden variables given the observed variables, called the posterior probability distribution.

$$p(\phi_{1:K}, \Theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\phi_{1:K}, \Theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \quad \text{(formula2)}$$

The exact estimation can be computed by summing the joint distribution over every possible instantiation of the hidden structure (i.e., assigning each observed word to every possible topic), which is computationally unfeasible. Topic model uses two different algorithms to approximate the formula 3 by adapting an alternative distribution over the latent topic structure to be close to the true posterior: sampling algorithms and variational algorithms. Sampling algorithms are attempting to collect samples from the posterior to approximate it with an empirical distribution. Gibbs sampling is the most common sampling algorithm [Geman et al, 1984]. It consists of a definition of a Markov chain on the hidden topic variables for a corpus. The process is iterated multiple times to collect samples from the posterior and then approximate the distribution with the collected samples.

In [Griffiths et al, 2004**]** the authors estimate the topic distributions over the vocabulary φ and document distributions over topics θ as follows:

$$\varphi_{kw} = \frac{C_{Kw}^{\phi} + \beta}{\sum_W C_{Kw}^{\phi} + W\beta} \quad \text{(formula3)}$$

$$\theta_{dk} = \frac{C_{dk}^{\theta} + \beta}{\sum_K C_{dk}^{\theta} + W\beta} \quad \text{(formula4)}$$

where $C^{\varphi}{}_{kw}$ maintains a count of all topic word assignments $C^{\theta}{}_{dk}$ counts the document topic assignments, and α and β are the hyper-parameters for the Dirichlet priors, serving as smoothing parameters for the counts. Variational methods are a deterministic alternative to sampling algorithms that approximate the probability of the posterior through optimization [Blei et al, 2003] .They posit a parameterized family of distributions over the hidden variables and then find the member of that family that is closest to the posterior.

3.2. Global Architecture

Figure.. highlights the general overview of the proposed system.

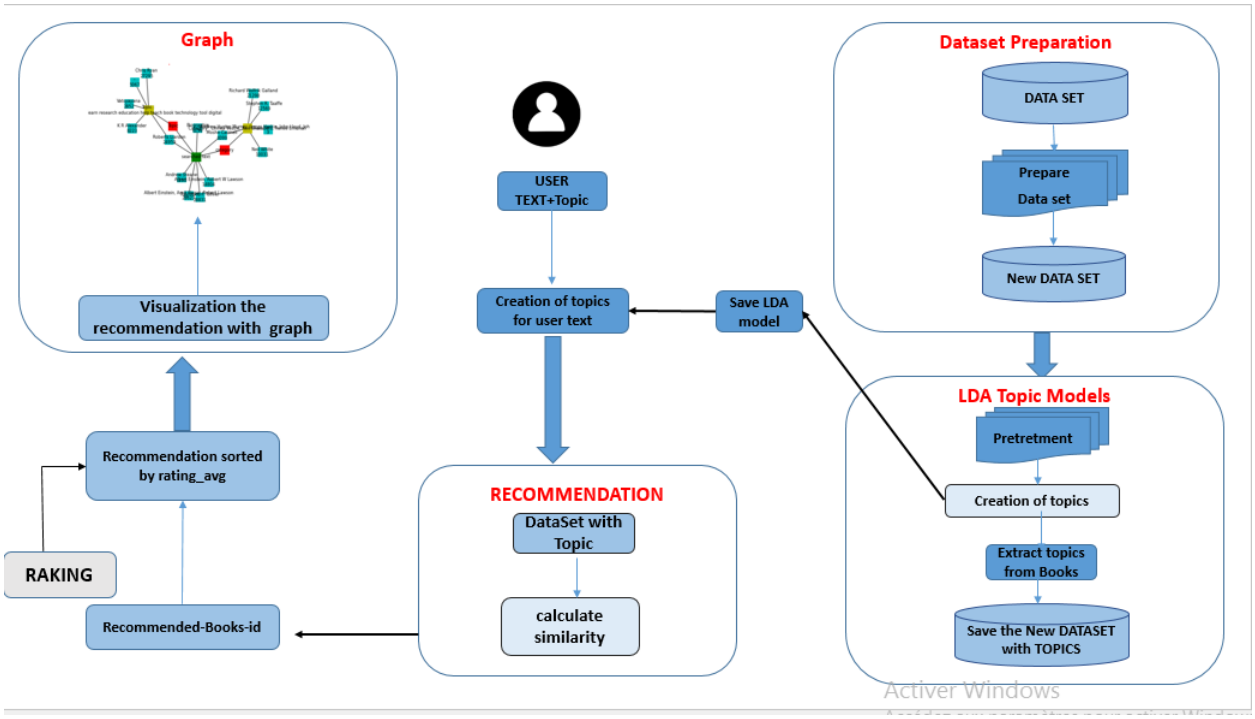


Figure5 Global architecture of the Proposed Book Recommender System.

3.3. Description of the Proposed Architecture

The objective fixed is to recommand a list of relevant books from a library based on input target text given by the user . To achive this goal we have proposed an hybrid approach for book recommender system we have first use an a content based technique. For this purpose the Latent Dirichlet Allocation (LDA) method have been exploited for books topic modelling. After the rating of books is used to rank resulted recommandation of the previous step, Finally, we have exploited a graph technique to visualise the final resulted recommandation associated with topics, which is a help tool for the user to dinstinguish the books relevance to topics.

We address this task of recommending books by means of the sub-tasks:

3.3.1. Data Preparation

The first step preceed the treatment process. It consists of cleaning the data so that it may be used later in the model. This task remove rows and columns with empty fields (Null/NaN values) and to focus only on the text data from each book, we also drop unusefull data.

3.3.2. Data Preprocessing :

In this stage of data preprocessing will perform the following steps:

- ✓ Recovery of the corpus.
- ✓ The first step is to retrieve the text. There are various methods for retrieving text, such as utilizing
- ✓ Tokenization: Split the text into sentences and the sentences into words.
- ✓ The normalization and creation of the dictionary, which allows for the omission of significant details at the local level (ponctuation, majuscules, conjugaison, etc.).
- ✓ Words are lemmatized and stemmed.

A. Corpus recovery

The first step is to retrieve the text. There are various methods for retrieving text, as scraping or downloading text files, for example.

B. Tokenization First and foremost, one must examine the vocabularies used in each book abstract and title. The text description is decomposed into word “Token” can be performed on them. The act of attempting to harmonize tokens is referred to as "normalization."

C. Normalization

- **Remove stop words**

  The first manipulation often performed in word processing is the removal of what are called stopwords in English. These are the very common words in the language studied ("and ", "will", "the ") which do not provide any informative value for the understanding of the "meaning" of a document and corpus. They are very frequent and slow down our work, so we want to remove them.

  The NLTK library includes a default set of stopwords in various languages, including English. However, we will approach it in a different way: we will eliminate the most frequently occurring terms from the corpus, assuming that they are part of the common lexicon and do not give any information. Then we will remove the stopwords given by NLTK.

- Words that have fewer than three characters are removed.
- Lowercase the ords and remove ponctuation.
-

D. Lemmatization or rootization

The "lemmatization" technique involves expressing words in their canonical form. It will be the infinitive of a verb, for example. It is singular masculine for a noun.,words in third person are changed to first person and verbs in past and future tenses are changed into present. The goal is to retain just the meanings of the words used in the corpus.

Another procedure that achieves a similar effect is known as rooting or stemming. This entails retaining only the root of the words examined, so, words are reduced to their root form. . The goal is to eliminate the suffixes and prefixes of the words, leaving simply their origin. It is a simpler and speedier procedure than lemmatization since the words are basically shortened, as opposed to lemmatization.

To verify whether the preprocessing, we will make a word cloud using the word-cloud package to et a visual representation of most common words. It is key to understand the data and ensuring we are on the right track, and if any, more preprocessing is necessary before training the model.

After the cleaning and preprocessing phase, we obtain a dictionary of words representing a new database of books with the number of occurrences of each word. We therefore represent each text by what is called a bag-of-words, which corresponds to the set of words that the text contains. In practice, this can be done by a frequency vector of appearance of the different words used. A classic bag-of-words representation will therefore be one in which we represent each text by a vector of the size of the vocabulary and we use the matrix composed of all of these N items that form the corpus as input to our algorithm.

2.3.2. Training the LDA Topic Models

In this stage several values of topics number K is fixed and LDA topic models are created and saved , where each topic is a combination of keywords. After that, cosine similarity is comuted to find the best number of topics. After that in every book from dataset associated topics are discovered using previously trained models. Therefore, a new dataset with topics is created.

Similarity Matrix Calculation

A. Term Frequency-Inverse Document Frequency (TF-IDF)

The TF-IDF weighting approach is widely used in information retrieval. Term frequency (TF) of a term t is the number of times it occurs in document d. A document in this context is all Inverse document frequency (formula4) helps distinguish the terms that are specific to a user/document.

$$\text{IDFt} = \log \frac{\text{N}}{\text{DFT}} \qquad \text{(formula4)}$$

N is the number of users and DFT is the number of documents where term t occurs. (formula5) defines the tf-idf weight of term t in document d.

$$\textbf{Tf-idf} = \textbf{tft,d} * \textbf{idft} \qquad \text{(formula5)}$$

B. Cosine Similarity

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words Let ***x*** and ***y*** be two vectors for comparison. Using the cosine measure as a similarity (formula6), we have

$$sim(x, y) = \frac{x*y}{|x|*|y|} \text{ (formula6)}$$

Where ||***x***|| is the Euclidean norm of vector x=(x1, x2… xp), defined

As: $\sqrt{x}_1^2 + \sqrt{x}_2^2 + \sqrt{x}_3^2 \ldots\ldots + \sqrt{x}_p^2$ Conceptually, it is the length of the vector. Similarly, ||***y***|| is the Euclidean norm of vector ***y***. The measure computes the cosine of the angle between vectors ***x*** and ***y***. A cosine value of zero means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. The closer the cosine value to one, the smaller the angle and the greater the match between vectors.

C. Books TF-IDF models

Descriptions of books are represented by the feature vectors through the TF-IDF model to analyses their content. TF-IDF model will give us the values on which we can calculate similarities. Therefore, cosine similarity matrix covering all the books is computed.

3.3.3.Recommendation generation

Once models are created, to generate recommendation, user enter a text request, the text is analyzed and associated with topics. After that, the topics user text are matched with books in augmented dataset. The system will process by calculating the similarities to make tewenty recommendations.

3.3.4.Raking Phase

Based on the resulted recommendation obtained from the content based technique, we applied in this second stage the second technique witch is a rating-based recommendation method. The purpose in this stage is to rank resulted recommendation. Therefore, based on average rating for

each book a classification is made over the books recommended in the previous phase. Thus top(N) example (N=7) are presented to the user.

3.3.5.Recommendation visualization and test

Times New Romangraph, three types of nodes are used, which are recommended books, topics and categories. The links in this graph represent the association between books and topics and also books and categories. Thus, from the resulted graph, user can localize the more relevant recommended book. He will eventually proceed to test it by consulting its description.

11.Conclusion

In this chapter, we discuss the architecture of our book recommendation system and the different steps to model. Which, is theoretically detailed just to show the functions, formulas and development tools used. The practice was based on the theory that has already been explained.

Chapter 3
Implementation and Experimentation

1. Introduction

In this Chapter, we first present the development tools used in implementing the proposed system, then we explain the steps of the realisation of books recommender system proposed in this study . The prototype is conducted to verify the effectiveness of our approach.

2. Development Environment and Tools

Two types of platforms are to be presented here, at the level of the hardware platform, we will present the machine on which we have built and test our system, with a description of the hardware configuration of the computer used during development. A software platform represents the tools and programming languages.

2.1 Hardware platform

To build our system, we used the following hardware configuration

- ✓ **Acer PC**: Processor (Intel(R) Core ™ i3 CPU M 370 @ 2.40GHz), RAM (3.00 Go).
- ✓ **Operating System**: Windows 7 Professional 32-bits.

2.2 Software platform

In this part we will briefly outline the development environment. We will discuss the tools, programming languages and utilities.

A. Programming languages:

Python

Python is the most widely used open source programming language among computer scientists. This language has propelled itself to the forefront of infrastructure management, data analysis or software development. Indeed, among its qualities, Python allows developers to focus on what they do rather than how they do it. It freed developers from the form constraints that occupied their time with older languages. Thus, developing code with Python is faster than with other languages [W1]. The main uses of Python by developers are:

-Application programming

- Creation of web services

-Code generation

-Meta-programming.

Jupyter

Jupyter is a web application used to program in more than 40 programming languages, including Python, Julia, Ruby, R, or even Scala. Jupyter is an evolution of the I Python project. it allows you to make notebooks or notebooks, that is to say programs containing both markdown text and code in Julia, Python, R. These notebooks are used in data science to explore and analyze data [W2].

Microsoft Excel

Spreadsheet software from the Microsoft office suite developed and distributed by the publisher Microsoft. The most recent version is Excel 2019. It is intended to run on Microsoft Windows, Mac OS X, Android or Linux platforms (using Wine). The Excel software integrates functions of numerical calculation, graphical representation, analysis of data (including pivot table) and programming, which uses macros written in the VBA (Visual Basic for Applications) language which is common to other Microsoft Office software [W3].

Anaconda

is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and mac.[W4]

Kaggle

Kaggle is a web platform organizing competitions in data science. On this platform, companies propose data science problems and offer a prize to data logists achieving the best performance. Anthony Gold bloom founded the company in 2010 [W5].

3. Implementation Steps

3.1. DATASET

In order to perform the necessary tests for the proposed system assessment, we used the dataset « book depository ». It contains 37650 books and «ID» indexes metadata such as «id, Authors, title, description, categories, format, rating-avg, and rating-count», each book. As a part of this work we have extracted the necessary data from multiple files, and rows in a single file to simplify the evaluation.

File	DESCRIPTION
dataset.csv	**The file contains information about books** ✓ **Id: id of the book .it is unique for each book.** ✓ **Authors: name of author** ✓ **Title: contains the title of each book** ✓ **Description: description of each book** ✓ Rating-Avg : average of ratings for each book, ✓ Rating-count : number of ratings for each book.
authors.csv	Authors-id **:** it's unique for each Authors **Authors name: name of Authors**
categories.csv	Categories -id : it's unique for each category **Categories name: name of Category.**

TABLE: Description dataset

The figure(6) above shows the data that contains the dataset.csv before its treatment..

	id	authors	categories	title	description	format	rating-avg	rating-count
0	9781840189070	[1]	[214, 220, 237, 2646, 2647, 2659, 2660, 2679]	Soldier Five : The Real Truth About The Bravo ...	SOLDIER FIVE is an elite soldier's explosive m...	1	4.03	292
1	9781844547371	[2, 3]	[235, 3386]	Underbelly : The Gangland War	John Moran and Carl Williams were the two bigg...	1	3.6	335
2	9788416327867	[4]	[358, 2630, 360, 2632]	A Sir Phillip, Con Amor	Sir Phillip knew that Eloise Bridgerton was a ...	1	3.88	37211
3	9780571308996	[5, 6, 7, 8]	[377, 2978, 2980]	QI: The Third Book of General Ignorance	The Third Book of General Ignorance gathers t...	1	4.17	384
4	9780008352516	[9]	[2813, 2980]	The Hidden Power of F*cking Up	The Try Guys deliver their first book-an inspi...	2	3.9	5095

Figure 6: The dataset.csv before initial state

	author_id	author_name
0	9561	NaN
1	451324	# House Press
2	454250	# Petal Press
3	249724	#GARCIA MIGUELE
4	287710	#Worldlcass Media

Figure7: A view of Authors.csv file

	category_id	category_name
0	1998	.Net Programming
1	176	20th Century & Contemporary Classical Music
2	3291	20th Century & Contemporary Classical Music
3	2659	20th Century History: C 1900 To C 2000
4	2661	21st Century History: From C 2000 -

Figure8: A view of Categories.csv file

3.2. Loading Packages

To develop the application, several libraries are used., So we start by loading the following Python packages :

- ✓ **NLTK**: (http://www.nltk.org/install.html) is a leading platform for building Python programs to work with human language data.
- ✓ **Gensim**: (https://radimrehurek.com/gensim/install.html) a subject modeling package containing our LDA model.
- ✓ **stop_words**: (https://pypi.python.org/pypi/stop-words) Python library for managing common stop words in 39 languages.
- ✓ **Matplotlib**: (https://matplotlib.org) is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- ✓ **Seaborn**: (https://seaborn.pydata.org) is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- ✓ **NumPy**: (https://www.w3schools.com/python/numpy/numpy_intro.asp) is a Python library used for working with arrays.

- ✓ **Pandas**: (https://en.wikipedia.org/wiki/Pandas_(software)) is a software library written for the Python programming language _ for data manipulation and analysis.

- ✓ **Re: (Regular expression operations): (https://docs.python.org/3/library/re.html) This module provides regular expression matching operations similar to those found in Perl.**

- ✓ **Random:(https://www.tutorialsteacher.com/python/random-module) is a built-in module to generate the pseudo-random variables. It can be used perform some action randomly such as to get a random number, selecting a random elements from a list, shuffle elements randomly, etc.**

- ✓ **Pickle:(https://docs.python.org/3/library/pickle.html)the pickle module implements binary protocols for serializing and de-serializing a Python object structure.**

```
import gensim
from nltk.stem import WordNetLemmatizer
from stop_words import get_stop_words
import pickle
import pandas as pd
import re
```

Figure 9: Loading Packages

3.3. Data Preparation

The dataset of books (dataset.csv, authors.csv, categories.csv) are loaded with **read_csv()** function, after that **dropna**() function is used to remove rows and columns with empty fields(Null/NaN values). The file dataset.csv is merged with the file categories and authors which resulted the dataset shown in figure ().To focus only on the text data , title and description fields are merged .

	id	Authors	title	description	Categories	rating-avg	rating-count
0	9781840189070	Mike Coburn	Soldier Five : The Real Truth About The Bravo ...	SOLDIER FIVE is an elite soldier's explosive m...	Marie Clay, Colin Drury, Veronica Grace, Jenny...	4.03	292
1	9781844547371	John Silvester, Andrew Rule	Underbelly : The Gangland War	John Moran and Carl Williams were the two bigg...	Elise Foster, Dariusz Karnas	3.6	335
2	9788416327867	Julia Quinn	A Sir Phillip, Con Amor	Sir Phillip knew that Eloise Bridgerton was a ...	Scholastic, Gregory MacDonald, Pasquale de Luc...	3.88	37211
3	9780571308996	Andrew Hunter Murray, James Harkin, John Lloyd...	QI: The Third Book of General Ignorance	The Third Book of General Ignorance gathers t...	Linnea Vestre, Neil Chambers, Renee Emunah	4.17	384
4	9780008352516	The Try Guys	The Hidden Power of F*cking Up	The Try Guys deliver their first book-an inspi...	Inazo Nitob, Renee Emunah	3.9	5095

Figure10: the dataset after preparation

3.4. Pre-processing

Because the books are in textual form, some processing is necessary, such as changing upper case to lower case, eliminating punctuation marks, and deleting stop words. The retrieved words are then used to construct our vocabulary dictionary. This LDA technique makes use of Python libraries, namely the processing library. We will perform the following steps:

- **Tokenization**

```
#extracting words
df = pd.DataFrame(dDict)
display(df.head())
print('extracting words ...')
```

- **Normalization**

Remove stopwords, punctuation, and words less then three characters, the corresponding code is as follows :

```
#remove stop word
data = df['content'].values.tolist()
data = list(map(deleteNoWord,data))
letters = [chr(char) for char in range(97,123)]
other = ["one","two","id","image","photo","caption","send","pic","video","images", "just", "today","three","tree",
                  "story", "like","will","source","say","watch","play","duration","getty","newsletter", "go",
                  "can", "year", "make", "view", "read"]
```

```
def deleteNoWord(string):
    return re.sub('[^\w_\s-]', ' ',str(string))
```

- **Lemmatization**

```
#Lemmatization
for doc in data:
    wordnet_lemmatizer = WordNetLemmatizer()
    doc = doc.replace("\n"," ")
    doc = doc.replace("\'","")
    doc = doc.replace("\t","")
    doc = doc.replace("\"","")
    doc = gensim.utils.simple_preprocess(doc)

    doc = [wordnet_lemmatizer.lemmatize(word) for word in doc]
    doc = [wordnet_lemmatizer.lemmatize(word,pos='v') for word in doc]
    en_stop = get_stop_words('en')
    doc = [word for word in doc if not word in (en_stop+letters+other)]
    texts.append(doc)
```

3.5. Exploratory Analysis

To verify whether the preprocessing, we will make a word cloud using the wordcloud package to get a visual representation of most common words. It is key to understand the data and ensuring we are on the right track, and if any, more preprocessing is necessary before training the model.

```
# Import the wordcloud library
from wordcloud import WordCloud
# Join the different processed titles together.
long_string = ','.join(list( df['title'].values))
# Create a WordCloud object
wordcloud = WordCloud(background_color="white", max_words=5000, contour_width=3, contour_color='steelblue')
# Generate a word cloud
wordcloud.generate(long_string)
# Visualize the word cloud
wordcloud.to_image()
```

This code displays the cloud of words of the corpus treated, shown in figure (11)

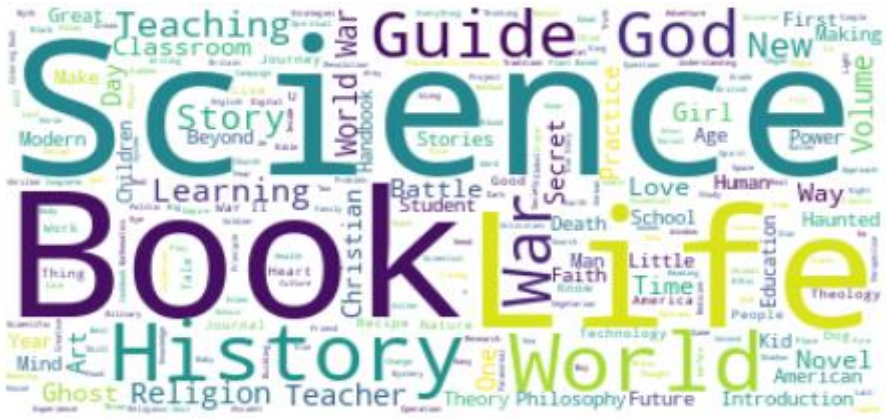


Figure11: Corpus cloud of words

3.6. Vocabulary dictionary

After the cleaning and preprocessing phase, we obtain a dictionary of words representing a new database of books with the number of occurrences of each word. We therefore represent each text by what is called a bag-of-words, which corresponds to the set of words that the text contains. In practice, this can be done by a frequency vector of appearance of the different words used. A classic bag-of-words representation will therefore be one in which we represent each text by a vector of the size of the vocabulary and we use the matrix composed of all of these N items that form the corpus as input to our algorithm.

Create a dictionary from 'processed_docs' containing the number of times a word appears in the training set.

```
def booksTopics(df,num_topics=50,isDf=True):
    from gensim import models
    import math
    lda_model = models.ldamodel.LdaModel.load('data/ldamodels_'+str(num_topics)+'.lda')
    id2word = pickle.load(open('data/pub_ldamodels_id2word.pkl','rb'))
    lda_topics = lda_model.show_topics(num_topics)
    lda_topics_words = ["".join([c if c.isalpha() else " " for c in topic[1]]).split() for topic in lda_topics]
    lda_topics_disp = [" ".join(topic) for i,topic in enumerate(lda_topics_words)]
    pickle.dump(lda_topics_disp,open('data/pub_lda_topics.pkl','wb'))
    book_topics = {}
```

The data is used for training and testing the proposed model.20% of the data are used for the algorithm test implemented and 80% are used for training

```
print('generating train and test files, please wait ...')
import random
train_set = random.sample(list(range(0,len(texts))),len(texts)-round(0.2*len(texts)))
test_set = [x for x in list(range(0,len(texts))) if x not in train_set]
```

3.7. Training LDA topic models

We train the model by changing the number of topics K. The generated models are saved.

```
print('train the model and save topic model ...')
from gensim import corpora, models
id2word = corpora.Dictionary(train_texts)
pickle.dump(id2word,open('data/pub_ldamodels_id2word.pkl','wb'))
corpus = [id2word.doc2bow(text) for text in train_texts]
ldamodels = {}
for i in range(20,100,20):
    random.seed(42)
    ldamodels[i] = models.ldamodel.LdaModel(corpus,num_topics=i,id2word=id2word)
    print('trained model for '+str(i)+' topics was saved to data/ldamodels_'+str(i)+'.lda')
    ldamodels[i].save('data/ldamodels_'+str(i)+'.lda')
```

```
train the model and save topic model ...
trained model for 20 topics was saved to data/ldamodels_20.lda
trained model for 40 topics was saved to data/ldamodels_40.lda
trained model for 60 topics was saved to data/ldamodels_60.lda
trained model for 80 topics was saved to data/ldamodels_80.lda
```

Figure 12: Training LDA topic models

3.8. Generating Book's Topics

We proceed to the generation of topics for each book in the dataset. The topics are saved in the dataset.

```
from gensim import models
print('get topics for all dataset ...')
lda_topics = {}
for i in range(20,100,20):
    lda_model = models.ldamodel.LdaModel.load('data/ldamodels_'+str(i)+'.lda')
    lda_topics_string = lda_model.show_topics(i)
    lda_topics[i] = ["".join([c if c.isalpha() else " " for c in topic[1]]).split() for topic in lda_topics_string]
pickle.dump(lda_topics,open('data/pub_lda_topics.pkl','wb'))

at = booksTopics(df)
print('dataset with topics is generating ...')
df = pd.read_csv('dbBooks.csv')#
data_dict = {'id': at.keys(),
                    'topics': at.values()}
dff = pd.DataFrame(data_dict)
df = pd.merge(df, dff, on='id')
display(df.head())
df.to_csv('bookDatasetTopics.csv')
print('dataset with topics was saved with Success')
```

```
get topics for all dataset ...
dataset with topics is generating ...
```

	Unnamed: 0	id	Authors	title	description	Categories	rating-avg	rating-count	topics
0	0	9781840189070	Mike Coburn	Soldier Five : The Real Truth About The Bravo ...	SOLDIER FIVE is an elite soldier's explosive m...	Marie Clay, Colin Drury, Veronica Grace, Jenny...	4.03	292	war world force fight german first military so...
1	1	9781844547371	John Silvester, Andrew Rule	Underbelly : The Gangland War	John Moran and Carl Williams were the two bigg...	Elise Foster, Dariusz Karnas	3.6	335	new time york novel author thriller bestsellin...
2	2	9788416327867	Julia Quinn	A Sir Phillip, Con Amor	Sir Phillip knew that Eloise Bridgerton was a ...	Scholastic, Gregory MacDonald, Pasquale de Luc...	3.88	37211	life love girl know find people time way come ...
3	3	9780571308996	Andrew Hunter Murray, James Harkin, John Lloyd...	QI: The Third Book of General Ignorance	The Third Book of General Ignorance gathers t...	Linnea Vestre, Neil Chambers, Renee Emunah	4.17	384	science human world new scientific philosophy ...

Figure13 : The new dataset augmented with topics.

3.9. Calculating similarities and generating recommendation

```
recommended_books_id = books_similarity_score.flatten().argsort()[::-1]
recommended_books_id_with_topic = books_similarity_with_topic.flatten().argsort()[::-1]
firstBook = recommended_books_id[0]
cat = df['Categories'].iloc[firstBook]
recNode = entredString+' '+cat
user_vector_cat = tfidf_matrix.transform([recNode])
books_similarity_with_category = cosine_similarity(book_tfidf_matrix, user_vector_cat)
recommended_books_id_with_category = books_similarity_with_category.flatten().argsort()[::-1]

final_recommended_books_id = [book_id for book_id in recommended_books_id][:NUM_RECOMMENDED_BOOKS]
final_recommended_books_id_with_topic = [book_id for book_id in recommended_books_id_with_topic][:FINAL_NUM_RECOMMENDED_BOOKS-2]
final_recommended_books_id_with_category = [book_id for book_id in recommended_books_id_with_category][:FINAL_NUM_RECOMMENDED_BO

print(final_recommended_books_id)
```

The output of the recommendation procedure is as follows:

	id	Authors	title	description	Categories	rating-avg	rating-count	topics
3	9780571308996	Andrew Hunter Murray, James Harkin, John Lloyd...	QI: The Third Book of General Ignorance	The Third Book of General Ignorance gathers t...	Linnea Vestre, Neil Chambers, Renee Emunah	4.17	384	history century work first world study early v...
12569	9780700619429	Stephen R. Taaffe	Marshall and His Generals : U.S. Army Commande...	General George C. Marshall, chief of staff of ...	Colin Drury, Frithjof Rodi, Franz Gamillscheg,...	3.81	64	war world air first force battle american navy...
6099	9781860942341	Moshe Carmeli	Group Theory And General Relativity: Represent...	This is the only book on the subject of group ...	Marc S. Kraus, Sunny Payne, Georges Fouron, Mi...	4.5	4	student learn teacher teach book use classroom...
14804	9781979997034	Albert Einstein, Robert W Lawson	Relativity - the Special and General Theory	Relativity: The Special and the General Theory...	Marcy Pavord, Georges Fouron, Meyerheiner, Gar...	4.19	18295	science human world scientific theory nature l...
21298	9781787392496	Richard Wolfrik Galland	Art of War Strategic Puzzles : Battlefield sce...	Art of War Strategic Puzzles provides the armc...	Frithjof Rodi, Rachel Morris	2.17	6	german army british military campaign operatio...
13828	9780094798502	General Sir Frank Kitson	Prince Rupert: Admiral and General at Sea	This text takes up the story of Prince Rupert ...	Colin Drury, Rose Elliot, Frithjof Rodi, Donna...	4	4	german army british military campaign operatio...
6159	9781599863719	Biographiq	General George S. Patton - Old Blood and Guts ...	General George S. Patton - Old Blood and Guts ...	Colin Drury, Kathy Sylva	4.33	3	german army british military campaign operatio...
15176	9781408851791	Rick Stroud	Kidnap in Crete : The True Story of the Abduct...	This is the story of how a small SOE unit led ...	Rena Salaman, Michael J. Benton, Franz Gamills...	3.61	87	war battle army fight unit tank operation worl...
			The Wonderful World of	This book provides a	Patrick Regan, Ana-Elba			child book help

Figure14 Recommendation without ranking

3.10. Ranking Phase

In the previous step 20 recommendation are provided and according to the rating_avg, we make a classification of the recommendations and we display 7 recommendation.

```
rdf = df.iloc[final_recommended_books_id]
display(rdf)
rdf = rdf.sort_values(by=['rating-avg'],ascending=False)
rdf = rdf.iloc[0:FINAL_NUM_RECOMMENDED_BOOKS]
final_recommended_books_id = rdf.index.values
rdf
```

Thus the list of top (7) recommended books

	id	Authors	title	description	Categories	rating-avg	rating-count	topics
7215	9780691026107	Albert Einstein, Anna Beck	The Collected Papers of Albert Einstein, Volum...	This volume presents Einstein's writings from ...	Maria Holmes, Vicky Howard, Georges Fouron, Me...	5	1	science human world scientific theory nature l...
6099	9781860942341	Moshe Carmeli	Group Theory And General Relativity: Represent...	This is the only book on the subject of group ...	Marc S. Kraus, Sunny Payne, Georges Fouron, Mi...	4.5	4	student learn teacher teach book use classroom...
6159	9781599863719	Biographiq	General George S. Patton - Old Blood and Guts ...	General George S. Patton - Old Blood and Guts ...	Colin Drury, Kathy Sylva	4.33	3	german army british military campaign operatio...
28831	9780764306785	Raymond F. Toliver	Fighter General: The Life of Adolf Galland: Th...	Adolf Galland began World War II in Poland, as...	Marie Clay, Michael Gill, Karen Schrier	4.29	24	war world air first force battle american navy...
1151	9780198789208	Andrew Steane	The Wonderful World of Relativity : A precise ...	This book provides a lively and visual introdu...	Patrick Regan, Ana-Elba Pavon, Uli Locher, Mey...	4.21	19	child book help young feel parent kid way lear...
14804	9781979997034	Albert Einstein, Robert W Lawson	Relativity - the Special and General Theory	Relativity: The Special and the General Theory...	Marcy Pavord, Georges Fouron, Meyerheiner, Gar...	4.19	18295	science human world scientific theory nature l...
28623	9780760759219	Albert Einstein, Amit Hagar, Robert Lawson	Relativity (Barnes & Noble Library of Essentia...	Albert Einstein's "Relativity: The Special and...	Vicky Howard, Meyerheiner	4.19	18165	science human world scientific theory nature l...

Figure15: Top(7) recommendation with raking

3.11. Visualizing recommendation graph

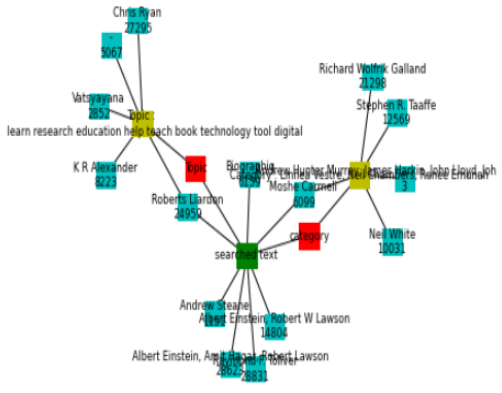
The resulted recommendation is displayed as a graph using networks package .

```
import networkx as nx
from matplotlib import pyplot as plt
recommandedBooksId = final_recommended_books_id
recommandedBooksTitle = [df['Authors'][bookId]+'\n'+str(bookId) for bookId in recommandedBooksId]
recommandedBooksTitleWithTopic = [df['Authors'][bookId]+'\n'+str(bookId) for bookId in final_recommende
recommandedBooksTitleWithCategory = [df['Authors'][bookId]+'\n'+str(bookId) for bookId in final_recomme
#G = nx.DiGraph()

G = nx.Graph()

node1 = 'searched text'
node2 = 'Topic'
node3 = 'category'
node4 = 'Topic : \n'+topic
node5 = 'Category : '+cat
```

The recommendation graph is illustrated in figure (16)




```
<Figure size 800x300 with 0 Axes>
```

```
Out[13]: 'display("Searched text : "+entredString)\n\nprint(\'Recommandation for "\'+entredString+\'" sorted by rating-avg\')\ndf.iloc[final_recommended_books_id]'
```

Figure 16: The recommendation graph

3.12. Test

Finally, from the recommendation graph the user can choose the book that seem more relevant He can enter the recommended book id and read the description as displayed below.

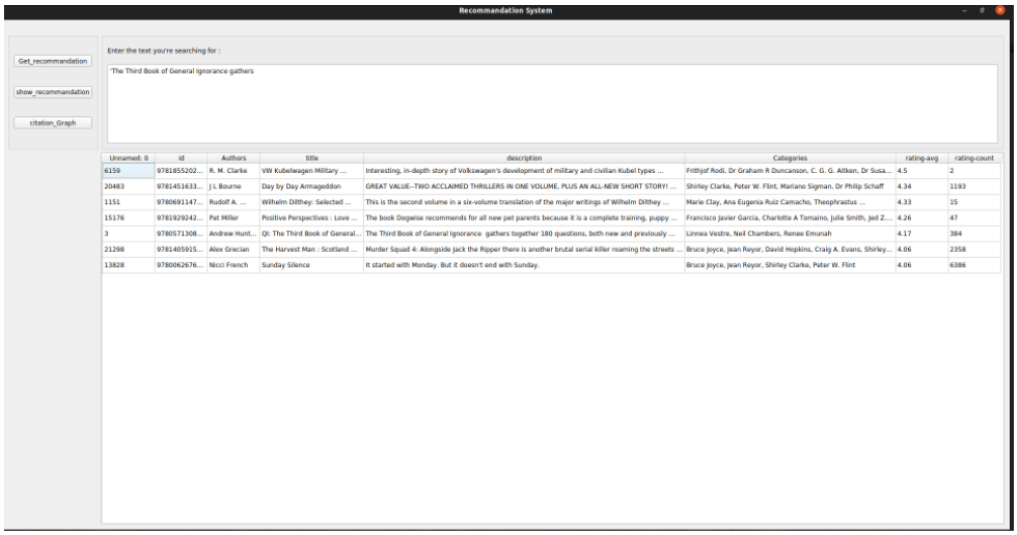
```
df.iloc[6099]['description']
```

```
"This is the only book on the subject of group theory and Einstein's theory of gravitation. It contains an extensive discussion
on general relativity from the viewpoint of group theory and gauge fields. It also puts together in one volume many scattered,
original works, on the use of group theory in general relativity theory.There are twelve chapters in the book. The first six ar
e devoted to rotation and Lorentz groups, and their representations. They include the spinor representation as well as the infi
nite-dimensional representations. The other six chapters deal with the application of groups -particularly the Lorentz and the
SL(2,C) groups - to the theory of general relativity. Each chapter is concluded with a set of problems.The topics covered range
from the fundamentals of general relativity theory, its formulation as an SL(2,C) gauge theory, to exact solutions of the Einst
ein gravitational field equations. The important Bondi-Metzner-Sachs group, and its representations, conclude the book. The ent
ire book is self-contained in both group theory and general relativity theory, and no prior knowledge of either is assumed.The
subject of this book constitutes a relevant link between field theoreticians and general relativity theoreticians, who usually
work rather independently of each other. The treatise is highly topical and of real interest to theoretical physicists, general
relativists and applied mathematicians. It is invaluable to graduate students and research workers in quantum field theory, gen
eral relativity and elementary particle theory."
```

Figure17: Description of recommended book

3.13. Application Interface

Our system's functionalities are accessible through a graphical user interface. In which, there is a field to input the user text request, and buttons to start recommendation and visualize it either in table form or in graph form, as shown in figure



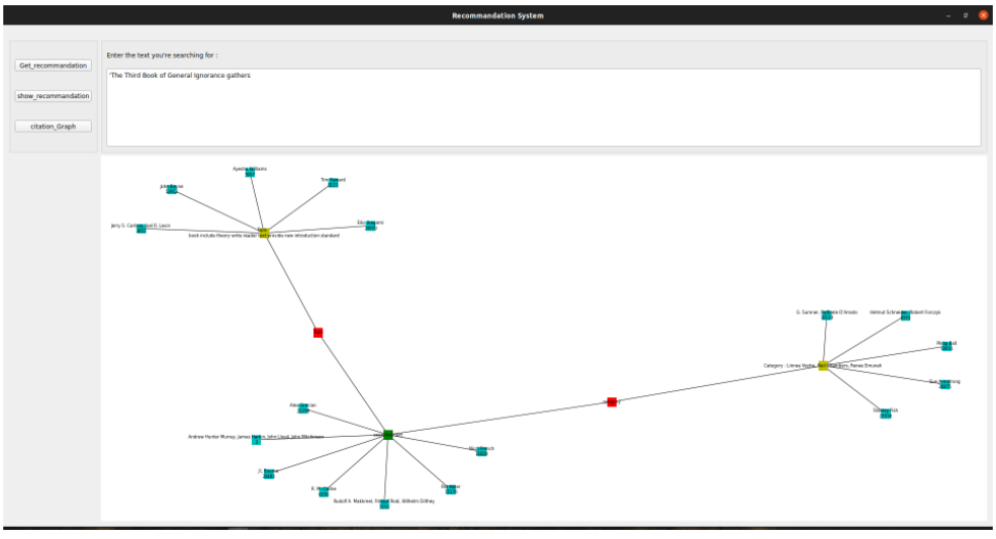


Figure18: The proposed Book Recommender System Interface

12.Experimentations and Evaluation

In addition to number of topics K , LDA model acts with two parameters alpha, and Beta :

- ✓ ***Alpha parameter*** *is Dirichlet prior concentration parameter that represents document-topic density — with a higher alpha, documents are assumed to be made up of more topics and result in more specific topic distribution per document.*
- ✓ ***Beta parameter*** *is the same prior concentration parameter that represents topic-word density — with high beta, topics are assumed to made of up most of the words and result in a more specific word distribution per topic.*

In this part, we did an evaluation for the LDA training model; we take a book from test-set when the model has not seen before we divide it into two parts (first part of the book, second part of the book). Therefore, we calculate the similarity between the two parts of the book and each time we change the values of alpha

- ✓ The smaller similarity (converges to 0) i.e. the LDA model is with good precision
- ✓ The similarity converges to 1, i.e. precision of the LDA model is reduced

Same books

	topicNum = 20	topicNum = 40	topicNum = 60	topicNum = 80
alpha = 0.01	0.649680	0.571280	0.536121	0.478735
alpha = 0.05	0.664167	0.612675	0.536407	0.509863
alpha = 0.1	0.612675	0.509863	0.520249	0.519605
alpha = 0.5	0.813362	0.753377	0.728058	0.687492
alpha = 1	0.753377	0.687492	0.924665	0.916721

random books

	topicNum = 20	topicNum = 40	topicNum = 60	topicNum = 80
alpha = 0.01	0.189180	0.146844	0.126809	0.098638
alpha = 0.05	0.184270	0.155009	0.113782	0.108820
alpha = 0.1	0.155009	0.108820	0.128120	0.115934
alpha = 0.5	0.338653	0.332134	0.312841	0.220017
alpha = 1	0.332134	0.220017	0.703558	0.699285

Note : smaller values are better

Figure 19: similarity of same books and random books

Experimentation 1

The figures of this experiment represent the variations of LDA training model evaluations as a function of similarity; giving values: alpha= 0.01, number topics= 20,40,60,80.

The different values of Number topics are represented on the y-axis while the x-axis is reserved for Average cosine similarity.

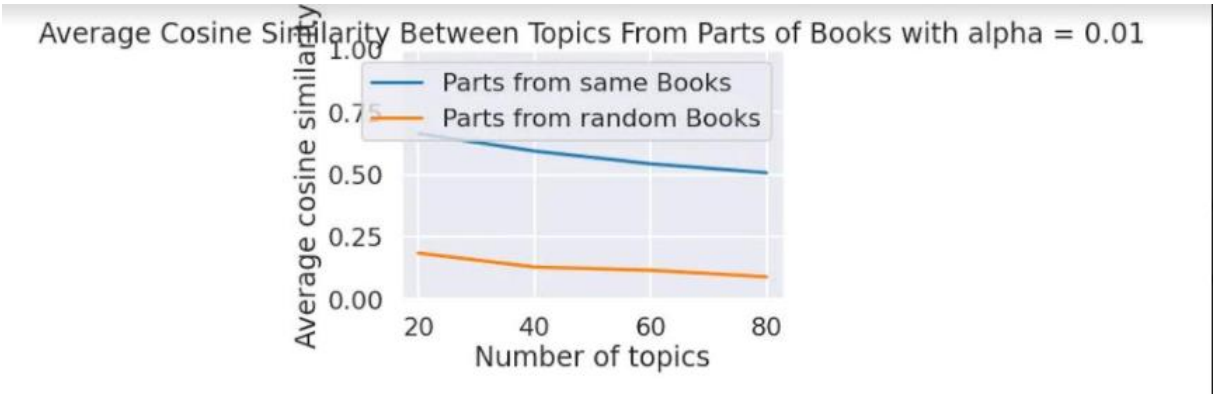


Figure20: Average cosine similarity between topics from parts of books with alpha = 0.01

Experimentation 2

The figures of this experiment represent the variations of LDA training model evaluations as a function of similarity; giving values: alpha= 0.05, number topics= 20,40,60,80.

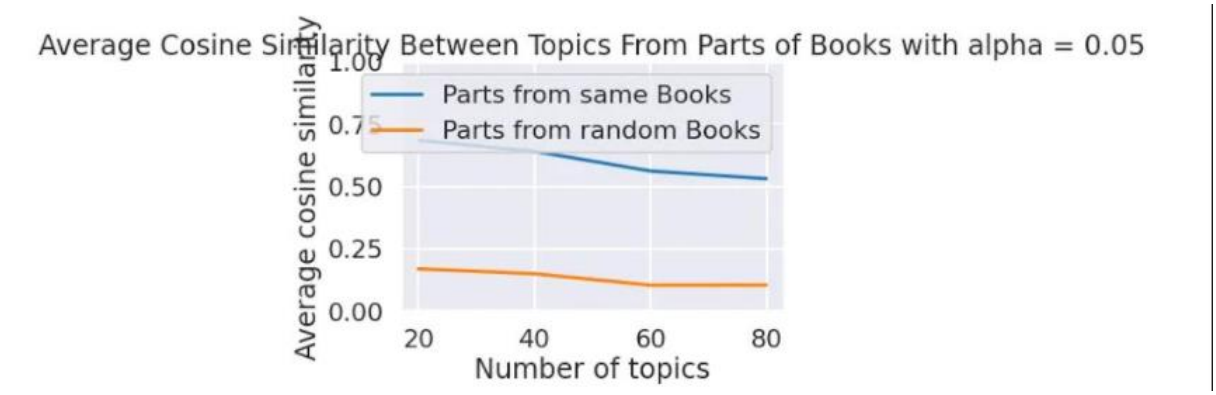


Figure21: Average cosine similarity between topics from parts of books with alpha = 0.05

Experimentation 3

The figures of this experiment represent the variations of LDA training model evaluations as a function of similarity; giving values: alpha= 0.1, number topics= 20,40,60,80.

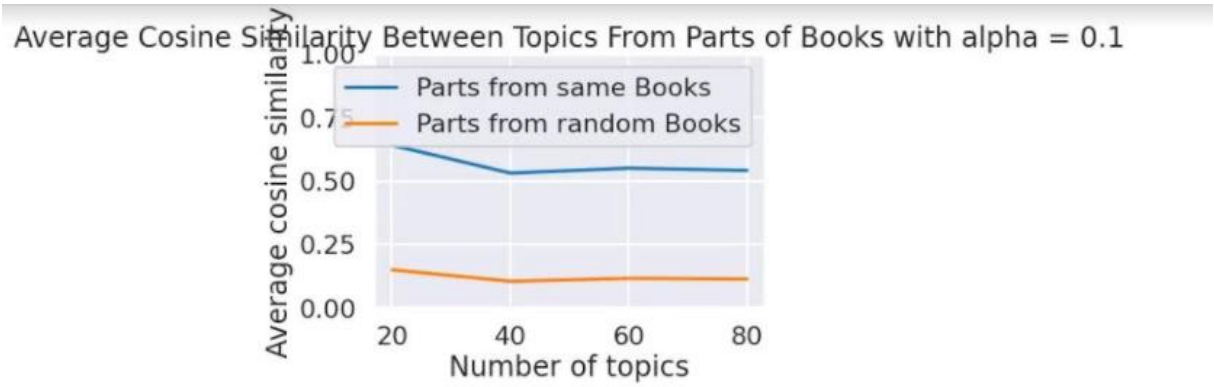


Figure22: Average cosine similarity between topics from parts of books with alpha = 0.1

Experimentation 4

The figures of this experiment represent the variations of LDA training model evaluations as a function of similarity; giving values: alpha= 0.5, number topics= 20,40,60,80.

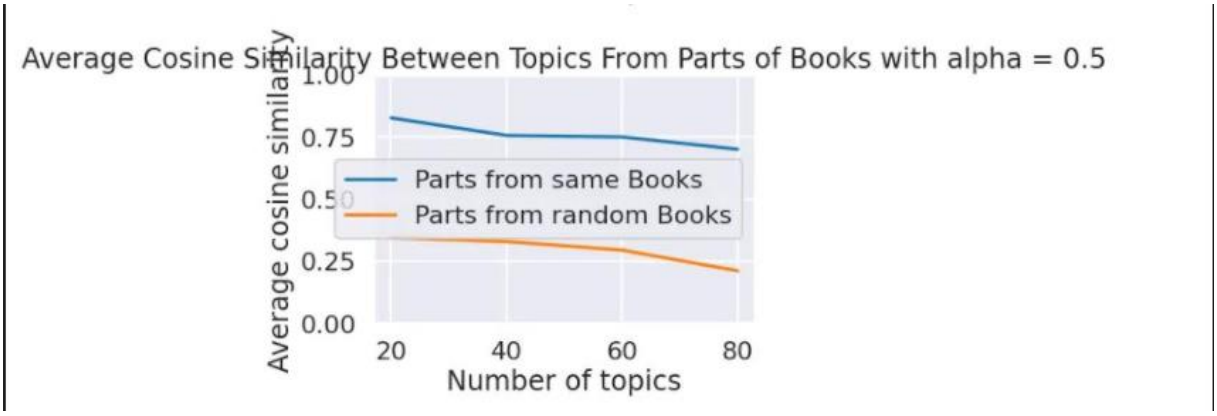


Figure23: Average cosine similarity between topics from parts of books with alpha = 0.5

Experimentation 5

The figures of this experiment represent the variations of LDA training model evaluations as a function of similarity; giving values: alpha= 1, number topics= 20,40,60,80.

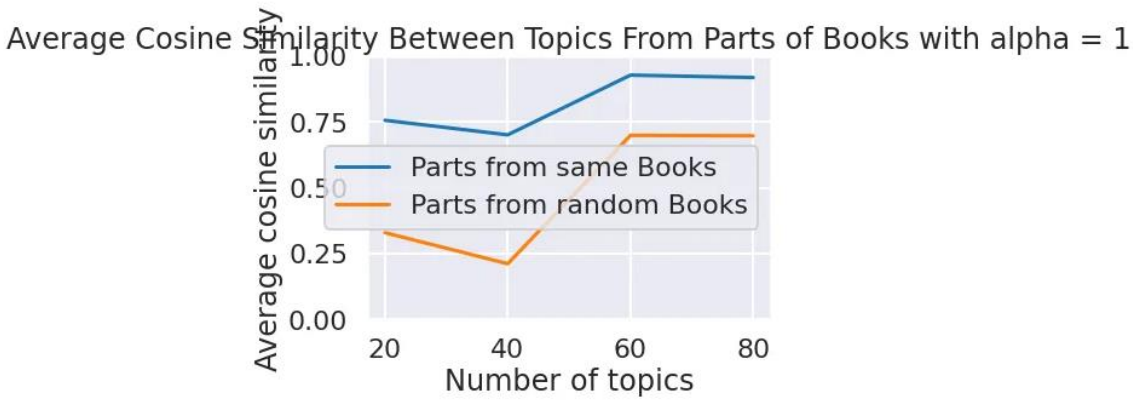


Figure24: Average cosine similarity between topics from parts of books with alpha = 1

5. Conclusion

In this chapter we have explained the steps followed to implement of the proposed book recommender system. We have also conducted exprementations and train LDA model for setting LDA parameters which provide the best results .

Conclusion and Perspectives

In this dissertation , we have proposed a hybrid recommendation system combining the LDA method with ranking and graphs techniques for the recommendation of books. LDA is used to find the latent semantic structure, the distribution of words on the latent topics and the mixture of the distributions of the latent topics, from the textual description of books The recommendation is done on two steps first cosine similarities between user request and books features are computed to find a list of candidate books. This list is the input of the second step which applied a ranking method to improve recommendation results by extracting the top N recommended books. At the end the recommendation is visualized with a graph which focus on books and topics to help user select the more relevant book and therefore display its description. A prototype is realized, and we have conducted an experimentation on a selected dataset which allowed us confirming the feasibility of our approach.

Perspectives:

As perspectives for future research, we plan to:

- ✓ Test the proposed model with other databases su.ch as Book-Crossing or Good-reads.
- ✓ Applying evaluation metrics to measure the performance of the realized system
- ✓ Consider contextual information in order to have a personalized recommendation
- ✓ combine the LDA approach with another approach. As a collaborative filtering based user rating.
- ✓ Exploit the potential of graph based methods using a specific algorithm , in order to obtain more accurate recommendation.

References

A. Bibliographical references

[B.M. Sarwarm et al. 2012] B.M. Sarwarm et al., "Analysis of Recommendation Algorithms for E-Commerce," ACM Conf. Electronic Commerce, ACM Press, 2000, pp.158-167

[GUR et al., 2013] Gurini, D. F., Gasparetti, F., Micarelli, A., & Sansonetti, G. (2013). A Sentiment-Based Approach to Twitter User Recommendation. *In RSWeb@ RecSys*.

[Griffths, 2004**]** Griffths, T. L. & Steyvers, M. Finding scientific topics. Proceedings of the National academy of Sciences, 101(suppl 1) :5228-5235, (2004).

[BUR 2002] Burke R., (2002): Hybrid recommender systems: Survey and experiments. In *User Modeling and User Adapted Interaction*, 12(4), pp. 331-370.

[Gadanho SC et al. 2007] Gadanho SC, Lhuillier N. Addressing uncertainty in implicit preferences. In: Proceedings of the 2007 ACM conference on Recommender Systems (RecSys '07). ACM, New York, NY, USA; 2007. p. 97–104.Google Scholar

[SHI et HAN, 2014] Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1), pp. 3.

[KEL et al, 2003] Kelly, D., & Teevan, J. (2003, September). Implicit feedback for inferring user preference: a bibliography. *In ACM SIGIR Forum* , Vol. 37(2), pp. 18-28. ACM.

[LEE et al., 14] Lee, W. J., Oh, K. J., Lim, C. G., & Choi, H. J. (2014). User profile extraction from Twitter for personalized news recommendation. *In ICACT'16,* pp. 779-783. IEEE.

[NGU et al., 2006a] Nguyen A., Denos N., Berrut C., (2006). Exploitation des données "disponibles à froid" pour améliorer le démarrage à froid dans les systèmes de filtrage d'information, *in INFORSID '06*, pp. 81-95.

[Oard DW, Kim J . 1998] Oard DW, Kim J. Implicit feedback for recommender systems. In: Proceedings of 5th DELOS workshop on filtering and collaborative filtering; 1998. p. 31–6.

[BOU 2005] Bouzghoub M., Kostadinov D., (2005). Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de profils, *CORIA'05*, 5, pp. 201-218, France.

[Gediminas Adomavicius.2011] Gediminas Adomavicius. "Context-AwareRecommender Systems" Systems Handbook, 2011, Recommender

[LEE et al., 2008] Lee, T. Q., Park, Y., & Park, Y. T. (2008). A time-based approach to effective recommender systems using implicit feedback. *ESA*, 34(4), pp. 3055-3062.

[REN et al.,20 09] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. *In Proc. of the 25th Conf. on UAI*. pp. 452-461.

[KOR et al., 2009] Koren Y., Bell R. and Volinsky C., (2009). Matrix Factorization Techniques for Recommender Systems. *IEEE Computer semi*, 42(8), pp. 30–37.

[MAATALLAH , 2015/2016] MAATALLAH , Une Technique Hybride pour les Systèmes e Recommandation, 2015/2016

[J. Buderet al .2012] J. Buder, C. Schwind Learning with personalized recommender systems: a

psychological view Comput Human Behav, 28 (1) (2012), pp. 207-216 ArticleDownload PDFView Record in ScopusGoogle Scholar

[Adamopoulos, Panagiotis, et al .2014] Adamopoulos, Panagiotis, and AlexanderTuzhilin. "On over-specialization andconcentration bias of recommendations :probabilistic neighborhood selection incollaborative filtering systems",Proceedingsof the 8th ACM Conference onRecommender systems - RecSys 14, 2014

[RVVSV Prasad et al.2012] RVVSV Prasad and V Valli Kumari, “A Categorial review of recommender systems”, 2012

[WEN et al. 2008] Weng L. T., Xu Y., Li Y., Nayak R., (2008). Exploiting Item Taxonomy for Solving Cold-Start Problem in Recommendation Making. In *the 20th IEEE Inter. Conf.*, Vol. 2, pp. 113-120.

[Rijsbergen C.J.V 1979] Rijsbergen C.J.V, (1979). Information Retrieval. *Second edition, Butterworks*.

[Sarwar et al., 2001] Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International Conference on World Wide Web, WWW ’01, pages 285–295, New York, NY, USA, 2001. ACM.

[P. C. Vaz et al. 2012a] P. C. Vaz, D. Martins de Matos, B. Martins, and P. Calado. Improving a Hybrid Literary Book Recommendation System through Author Ranking. In Proceedings of the 12th ACM/IEEECS Joint Conference on Digital Libraries, JCDL ’12, pages 387–388, New York, NY, USA, 2012a.

[P. C. Vaz et al. 2013] P. C. Vaz, R. Ribeiro, and D. M. de Matos. Understanding Temporal Dynamics of Ratings in the Book Recommendation Scenario. In Proceedings of the 2013 International Conference on Information Systems and Design of Communication

[Su and Khoshgoftaar, 2009] Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. Adv. in Artif. Intell., 2009:4:2–4:2, Jan. 2009.

[Koren, R et al , 2009] Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, Aug. 2009.

[Hu et al., 2008] Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In

2008 Eighth IEEE International Conference on Data Mining, pages 263–272, Dec 2008.
[Bai et al, 2019]. XIAOMEI BAI , MENGYANG WANG , IVAN LEE ,ZHUO YANG , XIANGJIE KONG ,Scientific Paper Recommendation: A Survey. Disponible en ligne

[Huang et al, 2002] Z. Huang, W. Chung, T.-H. Ong, and H. Chen ‘‘A graph-based recommender system for digital library,’’ in Proc. 2nd ACM/IEEE-CS Joint Conf. Digit. Libraries,2002, pp 65–73.
[P. C. Vaz et al. 2012b] Vaz, P.C., de Matos, D.M., & Martins, B. (2012b). Stylometric relevance-feedback towards a hybrid book recommendation algorithm. In *Proceedings of the fifth ACM workshop on research advances in large digital book repositories and complementary media, BooksOnline ’12* (pp. 13–16). New York: ACM.

[Mercy Milcah Y et al,2020] Mercy Milcah .Y , Moorthi K ,AI based Book Recommender System with Hybrid Approach, Vol. 9 Issue 02, February-2020, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181

[Ritu Rani et al, 2017] Ritu Rani*, Rahul Sahu. International journal of engineering sciences & research technology, “book recommendation using k-mean clustering and collaborative filtering” November, 2017

[Said, A et al, 2014] Said, A.; Bellogin, A.: Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In: Proceedings of The 8th ACM Conference on Recommender Systems RecSys’14. ACM, pp. 129–136, 2014.

[S. Vinodhini et al, 2014] S. Vinodhini, V. Rajalakshmi, B. Govindarajalu," Building Personalised Recommendation System With Big Data and Hadoop Mapreduce ", Vol. 3 Issue 4, April - 2014, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181.

[Jayanti Rathnavel et al, 2014] *Jayanti Rathnave and Kavita Kelkar* ,Personalized Book Recommendation System, *International Journal Of Engineering And Computer Science ISSN:2319-7242,Volume 6 Issue 4 April 2017, Page No. 21149-21153,Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i4.61*

[Shubham Gupta et al, 2020] Shubham Gupta, Yash Jain, Suchi Laad, Sudhanshu Khandelwal and Pritesh Saklecha, A Survey Report on Book Recommendation Techniques, Department of Computer science, Medicaps University Indore, India,July 31, 2020

[Monney et al, 2000] R.J.Monney and L.Roy, used content-based book recommending using learning for text categorization. “Proc. Of the Fifth ACM Conf.on Digital Libraries,p.195-204,2000.

[David M et al .2003] David M. Blei, Andrew Y. Ng, Michael I., Jordan, Latent Dirichlet

Allocation, Journalof Machine Learning Research (2003) 993-1022

[David M Blei, 2012] David — M Blei. Probabilistic topic models. Communications of the ACM, 55(4):77–84, 2012.

[Stuart Geman et al ,1984] — Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. IEEET ransactionson pattern analysis and machine intelligence, (6):721–741, 1984.

[Thomas L Griffiths et al ,2004] — Thomas L Griffiths and Mark Steyvers. Finding scientific topics. Proceedings of the National academy of Sciences, 101(suppl 1):5228–5235, 2004.

[DavidMBlei et al, 2003] — David MBlei, Andrew YNg,and MichaelIJordan.Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022, 2003.

[2019] — **End-To-End Topic Modeling in Python: Latent Dirichlet Allocation (LDA), 15-04-2019**

[Burke R,2002] — Burke R., "Hybrid Recommender Systems: Survey and Experiments", User modeling and user-adapted interaction, Vol. 12, N°. 4, pp. 331 - 370, 2002.

[TORI TOMPKINS,2020] — TORI TOMPKINS ,MAY Choosing the Right Recommendation Algorithm, MAY 13, 2020

B. Web References (Technical)

[W1] Définition of python
https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/

[W2] Définition of jupyter
https://fr.wikipedia.org/wiki/Jupyter

[W3] Définition of excel

https://en.wikipedia.org/wiki/Microsoft_Excel

[W4] Définition of Anaconda
https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)

[W5] Définition of Kaggle
https://fr.wikipedia.org/wiki/Kaggle

Abstract:

The work presented in this manuscript lies in the area of recommender systems which is often used to ride information overload. Several open source platforms have been made available for the development of RS. The objective of our work is to present the recommender systems, the main recommendation techniques, their problems and their solutions. We also exposed the topic modeling method and their applications in recommendation systems as well as the various research works in this field. We presented our recommendation system based on LDA (Latent Dirichlet Allocation) on a BOOK Depository database. The first step implements the LDA model from the descriptions of books ,we generate the recommendation ,then we integrate the raking method to improve our recommendation system, then we model the recommendation in the form of a graph. A prototype of this book recommender system is realized in order to evaluate the effectiveness of the proposed solution

Keywords : BOOK, LDA model, recommender systems, raking method

Résumé:

Le travail présenté dans ce manuscrit se situe dans le domaine des systèmes de recommandation qui est souvent utilisée pour sur monter la surcharge d'informations. Récemment, plusieurs plates-formes open source ont été disponibles pour le développement de RS. L'objectif de notre travail est de présenter les systèmes de recommandation, les principales techniques de recommandation, leurs problèmes et leurs solutions. Nous avons également exposé la méthode de topic modeling et leurs applications dans les systèmes de recommandation ainsi que les différents travaux de recherche dans ce domaine. On a présenté notre système de recommandation à base de LDA (Latent Dirichlet Allocation) sur une base de données BOOK .Depository.La première étape implémente le modèle LDA à partir des descriptions des livres la recommandation est ainsi générée , ensuite on a intégré la méthode de raking pour améliorer notre système de recommendation par la suite on a modélisé la recommandation sous forme d'un graphe. Un prototype de ce système de recommandation de livres est réalisé afin d'évaluer l'efficacité de la solution proposée.

Mots-clés : LIVRE, modèle LDA, systèmes de recommandation, méthode de ratissage

الملخص :

يكمن العمل المقدم في هذه المخطوطة في مجال أنظمة التوصية التي غالبًا ما تستخدم للتغلب على الحمل الزائد الهدف من عملنا هو تقديم .RS للمعلومات. في الآونة الأخيرة ، تم توفير العديد من المنصات مفتوحة المصدر لتطوير أنظمة التوصية وتقنيات التوصية الرئيسية ومشاكلها وحلولها. عرضنا أيضًا طريقة نمذجة الموضوع وتطبيقاتها في أنظمة التوصية بالإضافة إلى الأعمال البحثية المختلفة في هذا المجال. قدمنا نظام التوصية الخاص بنا استنادًا إلى يسمح النموذج المقترح بدمج المعلومات السياقية .BOOK في قاعدة بيانات (Latent Dirichlet تخصيص) LDA من أوصاف الكتب التي LDA حول المستخدمين والعناصر من أجل تحسين جودة التنبؤات. تنفذ الخطوة الأولى نموذج تم إنشاء التوصية بها ، ثم قمنا بدمج طريقة التجميع لتحسين نظام التوصية لدينا ، ثم قمنا بنمذجة التوصية في شكل رسم والرسم البياني للاقتباس. يتم إنتاج نموذج أولي لنظام توصية LDA بياني. أظهر هذا العمل كيفية الجمع بين طريقة .الكتاب هذا من أجل تقييم فعالية الحل المقترح

أظهر هذا العمل كيفية الجمع بين طريقة.الرسم البياني للاقتباس LDA التوصية في شكل رسم بياني للاقتباس

الكلمات المفتاحية: الكتاب ، نموذج LDA ، أنظمة التوصية ، طريقة التجميع