Toward a Real-Time Recommendation for Online Social Networks

By

Rania Albalawi

Thesis submitted to the University of Ottawa

in Partial Fulfillment of the Requirements for the degree of

Doctorate in Philosophy Degree in

Digital Transformation and Innovation

uOttawa

School of Electronic Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

(The most important reason that makes recommender systems so popular is the way they help giving the right information to the right user at a right time) (Wei, Moreau, 2003)

The Internet increases the demand for the development of commercial applications and services that can provide better shopping experiences for customers globally. It is full of information and knowledge sources that might confuse customers. This requires customers to spend additional time and effort when they are trying to find relevant information about specific topics or objects. Recommendation systems are considered to be an important method that solves this issue. Incorporating recommendation systems in online social networks led to a specific kind of recommendation system called social recommendation systems which have become popular with the global explosion in social media and online networks and they apply many prediction algorithms such as data mining techniques to address the problem of information overload and to analyze a vast amount of data. We believe that offering a real-time social recommendation system that can understand the real context of a user's conversation dynamically is essential to defining and recommending interesting objects at the ideal time.

In this thesis, we propose an architecture for a real-time social recommendation system that aims to improve word usage and understanding in social media platforms, advance the performance and accuracy of recommendations, and propose a possible solution to the user cold-start problem. Moreover, we aim to find out if the user's social context can be used as an input source to offer personalized and improved recommendations that will help users to find valuable items immediately, without interrupting their conversation flow. The suggested architecture works as a third-party social recommendation system that could be incorporated with other existing social networking sites (e.g. Facebook and Twitter). The novelty of our approach is the dynamic understanding of the user-generated content, achieved by detecting topics from the user's extracted dialogue and then matching them with an appropriate task as a recommendation. Topic extraction is done through a modified Latent Dirichlet Allocation topic modeling method. We also develop a social chat app as a proof of concept to validate our proposed architecture. The results of our proposed architecture offer promising gains in enhancing the real-time social recommendations.

Acknowledgement:

This thesis would not have been completed without the support of many people. I would like to dedicate this section to express my gratitude to all of them. First and foremost, I would like to thank Almighty ALLAH for HIS tremendous blessings and guidance. I could have never completed this work without the faith I have in HIM.

I would like to acknowledge my sincere appreciation and respect is intended for my supervisor Professor Tet Hin Yeap and my co-supervisor Professor Morad Benyoucef who have supported, guided, and encouraged me throughout my academic success. This thesis would not be a success without the assistance they have provided me. It was a wonderful experience to work with them and learn from them. Thank you to all of the professors at the University of Ottawa who sparked my passion in the field and provided me with the knowledge I needed. I am grateful for all my Ph.D. committee members, Professor Liam Peyton, Professor Umar Ruhi, Professor Bijan Raahemi, Professor Amiya Nayak, Professor Anis Yazidi, and Professor Iluju kiringa for their insightful comments and valuable suggestions.

I feel a deep sense of gratitude for my parents who formed part of my vision and taught me good things that really matter in life. Their infinite love and support have always been my strength. Their patience and sacrifice will remain my inspiration throughout my life. I am also very much grateful to all my brothers and sisters (Rami, Dr. Randa, Reema, Raad, Rowaida, Dr. Raneem, Rakan, Rayan, and Renad) for their constant inspiration and encouragement. Also, the most special thanks are reserved for Bader, who has always been there, through my ups and downs.

I would also like to extend my gratitude to all my friends and colleagues (Sahar, Leandra, Ellen, Abir, Noor, Sima, Laith, Arif, Sultan, and Ebrahimi) who helped me succeed especially my dearest Leandra who has been always unconditionally supportive and encouraging for me. Last but not least, I am grateful to my uncle Dr. Albalawi, Wafi for his support and guidance during my studies.

Table of Contents:

List of Figures:

List of Tables:

List of Abbreviations:

Acronym	Definition
API	Application Programming Interface
BoWs	Bag-of-Words
CB	Content-Based recommender
CF	Collaborative-Based recommender
CRSs	Conversational Recommender Systems
DGE	Graph Based Embedding
DSR	Design Science Research
DTA	Direct Targeted Advertising
HCI	Human Computer Interaction
HPM	Heuristic Pruning Method
HTTP	Hypertext Transfer Protocol
ISO	International Standards Organization
IR	Information Retrieval
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Indexing
ML	Machine Learning
MSM	Mobile Social Media
NLP	Natural Language Processing
NMF	Non-Negative Matrix Factorization
OSNs	Online Social Networks
PCA	Principal Component Analysis
POIs	Point of Interests
PLSA	Probabilistic Latent Semantic Analysis
RP	Random Projection
RSs	Recommender Systems
SM	Social Media
SMM	Social Media Marketing
SNA	Social Network Analysis
SNSs	Social Network Sites
SRSs	Social Recommender Systems
SVM	Support Vector Machines
TA	Threshold Algorithm
TF-IDF	Term Frequency/Inverse Document Frequency
TM	Topic Modeling
UGC	User Generated Content

Chapter 1 Introduction

Nowadays, people tend to rely heavily on the Internet in their daily social and commercial activities. The Internet helps to increase demand for the development of commercial applications and services that can provide better shopping experiences and commercial activities for customers around the world. The Internet is full of information and sources of knowledge that may confuse readers and cause them to spend additional time and effort in finding relevant information about certain topics of interest. Consequently, there is a need for more efficient methods and tools that can aid in detecting and analyzing the content of online social networks. Among systems using user-generated content as a source of data, recommendation systems are considered to be an important method to solve this issue. It is the most popular and productive way to facilitate users' access to information across the web that best meets their needs and preferences (Year. R, 2017).

Researchers in both academic and industrial fields have developed new methods in recommendation systems (**RSs**) (particularly for socialized, personalized, and real-time RSs) that are gaining attention in the e-commerce domain. This is done by learning information about users, such as their past purchase history. Generally, RSs can then provide users with suitable suggestions that match their interests to enhance their commercial activities. In turn, social media (**SM**) enables online communication and collaboration. It encourages consumers to post, like, and share comments, as well as share information about their experiences using a company's goods, services, and innovations. SM plays a significant role in our daily lives and in the activities of businesses, particularly regarding the promotion of online marketing activities and products. It is considered to be the most powerful platform on which to launch new products and foster viral marketing campaigns (Kaplan and Haenlein, 2011) and (Kaplan, 2012).

With the current rapid growth of RSs, developments in text analysis approaches are receiving significant interest from researchers Accordingly, social recommendation systems (**SRSs**) have become a hot topic. For example, by applying social network analysis methods and leveraging their social aspect, many vendors, like Amazon, have incorporated recommendation capabilities into their commerce servers and consider them to be an integral part of e-commerce (Peddy, 2003).

Despite all the interest, SRSs need more improvements, such as the ability to deal with information overload. Their function is to satisfy users by dynamically providing them with real-time personalized recommendations, content, and services in an efficient manner. Many approaches have focused on just implementing algorithms and methods that are only capable of specific features in SRSs. It has been proposed to analyze data in the social network domain using classification, clustering, and probabilistic techniques; however, such approaches are far from perfect and satisfactory to users, particularly mobile Internet users who are currently considered to be the major portion of total Internet users.

In this research, we will consider SRSs found in a wide range of applications within the online social networks (**OSNs**) domain that aim to provide useful recommendations at a particular time. However, many recommendations created by these systems do not always meet the customer's real-time requirements. They involve text mining that deals with analyzing unstructured and semi-structured text data. Consequently, researchers have been working to enhance the predictions generated by these systems to overcome several issues in SRSs, such as the cold-start problem, which we will provide a useful solution to.

1.1 Research Motivation

SRSs have become an important stream of research, as they can be used in numerous areas: within mobile apps or desktop applications, to enhance commercial activities, or to improve the user's satisfaction by recommending items, such as movies. In general, information about a user's preferences is a critical notion that guides our selections and activities. Offering the right items at the right time, only to interested users, is a challenging subject, since predicting items depends on several processes and attributes, such as the user's history, social relations, ratings, etc.

Researchers have been investigating many mechanisms and models in recent years to produce optimal recommendation results efficiently; however, some of those models did not prompt the user's preferences at the beginning of the interaction and users are often not satisfied with the initial recommendations (Bridge, 2006). SM platforms are currently one of the most attention-attracting aspects on the Internet to provide real-time recommendations to interested users such as real-time news and products. Several of the most popular SM platforms that store information as text written in natural language within the social network landscape (e.g., Twitter and Facebook) do not generally support real-time recommendations directly from the user's social media streams. Hence, it is more convenient to employ recommendation approaches that are similar to a human-human interaction where users can specify their preferences over an extended dialogue.

The problem of determining the real user's requirements and interests from his OSNs is not new. There have been many attempts to create approaches and data analysis techniques like the topic modeling method (**TM**), which was useful in predicting and recommending valuable objects to the user. Recent research has shown that different types of information flows in OSNs can improve the recommendation results, for instance by performing social network analysis (**SNA**) and benefiting from the social and time aspects. Many online vendors (e.g., Amazon) have incorporated recommendation capabilities into their commerce servers and consider them to be a part of e-commerce (Clayton, 2003). Such vendors rely on online reviews to acquire feedback, enrich customer satisfaction, and get useful comments from their customers to improve their processes. Text-based SRSs have become popular especially with the global explosion of SM. Many approaches have been proposed to mine and analyze the text in OSNs. Most are based on classification, clustering, and probabilistic techniques; however, current methods are far from perfect and fail to fully satisfy users.

This thesis does not try to define what it means to find a topic within a chat, but it tries to find out what topic is being discussed in the online user's conversation session. Our main motivation in this thesis is to design and develop an efficient SRS architecture that can automatically detect topics in the online chat then delivers proper recommendations that match the user's current interests and minimizes his/her decision-making time.

1.2 Motivating Scenarios

This section provides two motivating scenarios that demonstrate the benefits of the proposed SRS architecture. These scenarios will be used as examples of the implementation of the proposed real-time SRS that includes a prototype application to push notifications on the user's mobile phone. In the first scenario, let us assume that Sima and Leandra are two friends who are chatting through a mobile chat application, such as Facebook. Sima is an employee in a commercial company and was working for a few months on a project; she's tired and wants to take a break and travel for a week. However, she is swamped and does not have time to visit a travel agency or check online travel websites/apps to find a good deal on a vacation that matches her preferences. Consequently, the proposed system should extract the online generated-contents dynamically and determine the real user's requirement and interests by obtaining meaningful information that helps to detect the content's topic, not just the keywords, to suggest suitable recommendations then push notifications on the user's mobile phone like flights and travel destinations as required in Sima's situation.

In the second scenario, let us assume that Rami was sitting alone at home and he was chatting through a web application with his friend Bader who is interested in buying a car next year. After Rami exits his conversation window, he might receive some push-based recommendations (web-ads) based on his recent conversation, which are sent to him without his permission or a specific request. However, this kind of recommendation will not satisfy Rami as it is a waste of his time. Consequently, the proposed system should not send any recommendation to the user until he has agreed to receive content-related suggestions after determining his real interests.

Our research is motivated by the following insights we gathered from the literature:

- Many previous studies applied real-time SRS by using various approaches, such as defining an individual preference from other group members' preferences. Nevertheless, the user's actual and current requirements are rarely considered, which leads to unusable recommendations that are not accepted by users.
- Previous studies developed numerous text mining methods and natural language processing (**NLP**) techniques to predict users' keywords, interests, and priorities, but they mainly based it on their importance, word frequency, topical similarity, and rating; however, extracting keywords based on the user's online content is a challenging subject and needs further investigation as many previous studies consider offline data such as documents to provide more personalized recommendations. It is essential to mention that we are dealing with chat as an online input data source (informal and short dynamic sentences).
- Several studies have proposed solving the user/item cold-start problem by utilizing data from users' OSNs; however, current methods are still far from perfect and fail to satisfy users, particularly mobile users who are considered a major part of Internet users.

In this research, we develop a social chat app as a proof of concept to validate our proposed real-time SRS architecture. In our research we considers the topic extraction techniques in dynamic social interactions to enhance SRSs recommendations, which boosts users' satisfaction with their online activity involvement.

1.3 Research Questions

The literature surrounding SRSs deals with many issues and their solutions. Most of these solutions have been considered the user information and generated data in OSNs like the user's social information feature to recommend suitable items. However, to date, there have been limited works that aim to enhance SRSs based on online user-generated content (**UGC**). Following is the fundamental research question of our research which is,

- ***How to derive a real-time social recommendation from the user's online social network platforms and apps?***

In order to achieve this, we have address the following essential research questions:

- **What are the primary features and criteria in designing the main architecture of an efficient real-time social recommendation system?**
- **How does the topic modeling method contribute to the social recommendation system to recommend related tasks based on the user's context in a real-time manner?**

1.4 Research Objectives

We argue that information on OSNs, chat apps, in particular, can be considered a new, rich source of data that can be used to suggest personalized recommendations to users in real-time. Hence, we aim to propose a conceptual architecture based on users' online generated content to deliver suitable social and personalized recommendations.

To achieve this goal, we need to satisfy the following objectives:

- Design and build an architecture for a real-time SRS, called ChatWithRec, that aims to automatically detect topics from the user's online chat, then deliver suitable recommendations that match the user's current interest and minimize his/her decision-making time.
- Apply text mining and NLP approaches to social media content recommendations by analyzing unstructured and semi-structured text. Without a doubt, employing techniques that are similar to human-human interaction is more convenient, whereby users can specify their preferences for overextended content; hence, there is a need for effective methods and tools that can aid in detecting and analyzing OSN content, particularly for those using online UGC as a source of data.
- Enhance conversation-related tasks by defining user requirements and needs in real-time without considering any previous activities, such as rating items. We aim to help users find valuable items immediately, which will encourage them to participate more in conversations to generate appropriate recommendations for them dynamically without disrupting their conversation flow.
- Suggest a possible solution to the user cold-start problem, described as the difficulty in bootstrapping the recommendation system for new users.
- Develop a chat app as a proof of concept of our proposed architecture to collect streaming data from a user's OSNs.
- Apply several statistical metrics to evaluate and analyze the performance of the proposed real-time SRS architecture.

1.5 Research Contributions

The central intention of this thesis is to design, develop, and evaluate a real-time SRS architecture that aims to improve the word usage and understanding in OSN platforms by defining the user's current context and topic based on the user's online social streaming data to outline their definite requirements and then suggest more suitable and personalized recommendations automatically to interested users, such as advertisements, traveling plan, e-Commerce recommendations, and other conversation-related tasks.

The thesis contributions can be summarized by the following:

- Propose a real-time SRS that is able to integrate many subsystems to propose a well-organized and advanced system for OSN users.
- Propose a real-time SRS that displays recommendations based on analyzing the user's actual requirements and needs from their social online generated content, which takes into account the user's preferences, as well as personal and temporal contexts.
- Design and implement a conceptual architecture that can divide all the system's functions equally between the server and mobile app. Thus, the developed app will work as a complex app that we can rely on if the server goes down.
- Propose a real-time SRS that gives users full control over the system usage; hence, a user's generated content will be collected and analyzed to define relevant tasks based on their agreement and preference to avoid pushing irrelevant recommendations. Also, users can activate/deactivate the system service anytime and anywhere.
- Propose a real-time SRS that encourages users to engage more in chatting to narrow their options, and assists them in finding valuable tasks immediately without interrupting their conversation flow. This will ensure the usability of our system; therefore, the more sentences the user writes in a conversation, the more accurate the recommendations will be.
- Propose a real-time SRS that can adopt many topic modeling algorithms to enhance the system's understanding of users' conversations. It is worth mentioning that we are going to deal with chats as an online input source.
- Propose a real-time SRS that can provide a solution to the user cold-start problem. There is no need to obtain a user's tags, likes, or friends' preferences to predict their next recommendation through our proposed RS.

1.6 Scholarly Achievements

- **Rania Albalawi**, Tet Hin Yeap, and Morad Benyoucef, "Toward A Real-time Social Recommendation System," MEDES'19, November 12-14, Limassol, Cyprus, (2019).

- **Rania Albalawi**, Tet Hin Yeap, "ChatWithRec: Toward A Real-time Conversational Recommender System". ICCSMLBD, 17th-18th October. New York, USA. (2019). (**Excellent Paper Award**).
- **Rania Albalawi**, Tet Hin Yeap, and Morad Benyoucef, "Using Topic Modeling Methods for Short Text Data: A Comparative Analysis". A Journal in the Frontiers in Artificial Intelligence- Machine Learning and Artificial Intelligence. DOI: 10.3389/frai.2020.00042. Manuscript Id: 539160. (2020).
- **Rania Albalawi**, Tet Hin Yeap, and Morad Benyoucef. "Evaluating an Interactive Architecture for Real-Time Social Recommendation System". The 4th International Conference on Software Engineering and Information Management (ICSIM 2021). Yokohama, Japan. (January 16-18, 2021).

1.7 Thesis Organization

The thesis is organized as follow:

- **Chapter 2** presents the current work of the RSs subject, specifically the SRSs, and details its analytical techniques and methods. We will describe the most important features in an SRS to make it more efficient and valuable for users. Besides, we will give a general overview of the NLP, precisely the TM method.
- **Chapter 3** discusses the methodology we followed in this work, including both of the literature review and design science research methodologies.
- **Chapter 4** introduces the main idea of the proposed ChatWithRec real-time SRS, including its architecture and components.
- **Chapter 5** discusses the ChatWithRec system design and implementation details.
- **Chapter 6** presents the evaluation of the ChatWithRec system including its usability, user satisfaction, and quality of recommendations.
- **Chapter 7** concludes the thesis by summarizing our work in the field of real-time SRSs and suggests several avenues for future work.

Chapter 2 Background and Related Works

During the last few years, there has been a great demand for social recommendation systems and services that can provide the best shopping experiences and commercial services for customers. Those systems were improved by incorporating the social media paradigm that encouraged consumers to make purchases, track their activities, and map their social networks. Social media can help to enhance customers' shopping behavior as well as improve their shopping experience. It also enhances the business environment, especially for marketing companies who want to interact more with their consumers. This interaction is performed through data collection to analyze consumers' shopping experience through data mining. This allows marketing companies to solicit and receive more feedback so they can understand the consumers' buying habits as they pertains to their products and services, which in turn allows businesses to review, promote, and enhance their consumers' online business experience.

Many business enterprises use online reviews to acquire feedback and enrich customer satisfaction. Some examples are eBay and Amazon, companies that can easily access some of their users' information like their age, activity, gender, location, and time spent on their websites looking for particular products. Consequently, there is a need to extract useful and hidden information from numerous online sources that are stored as text, written in a natural language in the OSN platforms such as Twitter, LinkedIn, Instagram, and Facebook. SM such as OSNs can present many challenges, including, scalability, sparsity, data noise, the cold-start problem, and informal words problems which researchers have recently tried to solve by using NLP methods (Farzindar and Inkpen, 2015). The work conducted by Singh (2015) presented some major issues and challenges of user opinion mining, which is available on many social websites like blogs, Facebook, etc. and which has a substantial impact on users' decision-making. Furthermore, Wei et al (2017) integrated both the collaborative filtering (**CF**) approach and machine learning (**ML**) algorithms to tackle the cold-start problem which improved overall recommendation accuracy. However, Their approach focused only on time contexts.

We conducted an extensive background study to investigate the areas of research that need further exploring by identifying the main gaps in the literature and focusing on the contribution to this area. The purpose of this study is to provide a better understanding of the trend of SRSs and to determine how NLP and topic modeling benefit the commercial environment. We will begin our literature review by presenting a general overview of RSs, their background, and their main types and categories, focusing specifically on SRSs. Then we will continue with an overview of NLP and the topic modeling method. We will include a comprehensive review of many previous studies in our target domain and a brief discussion that summarizes the main findings.

2.1 Recommendation Systems RSs Background

The subject of OSNs is very popular. They give everyone the ability to communicate with others and share posts in the form of text, links, images, and videos. People can generate content easily and post anything they want in their accounts, which are considered to be the most significant features of OSNs today. Conversely, one of the most significant problems these sites face is information overload, which causes users to spend more time and effort to find relevant content that matches their interests among the massive amount of information available. Another issue that RSs suffer from is the cold-start problem. Many recent studies have employed online and offline UGC from OSNs to solve it; however, this issue is still unsolved, as many RSs predict users' preferences based mostly on their offline data.

Many studies describe RSs based on their use. Other studies define RSs as information filtering tools that deal with the problem of information saturation by filtering fundamental information fragments out of the vast amount of continuously changing data according to the user's preference, his/her interests, and/or noticed viewpoint about an item (Konstan. J, 2012; Pan, 2010). Also, (Sohail, 2017) defined that an RS determines the user's need, information, and preference and adjusts it to identify appropriate recommendations. Figure 2.1 shows a general diagram of a general RS as stated in (Reshma, 2020). From the e-commerce perspective, an RS is defined as a tool that supports users who want to search based on their relevant interests and preferences (Schafer, 1999). In general, RSs main functions are helping users to find high-quality and attractive items, enhancing the users' shopping experience, reducing the time wasted on browsing repeated or fake information, and recommending items that users need.

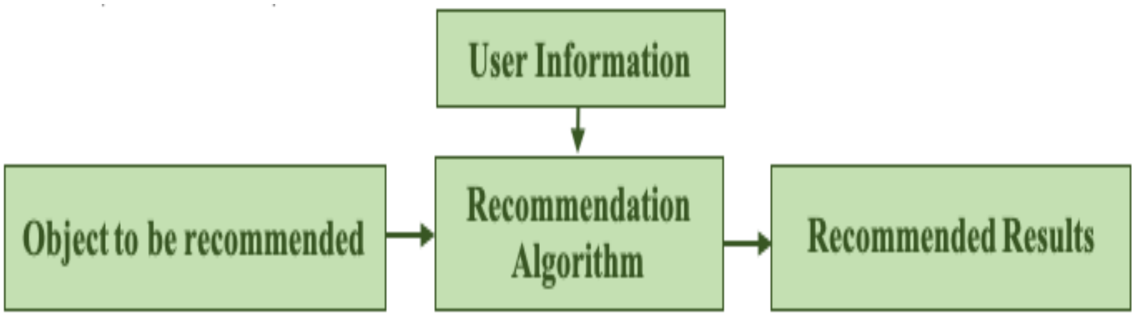


Figure 2.1: General Recommendation System (Reshma, 2020).

2.1.1 Classification of RSs

SRSs combine ideas from several research domains, including user modeling, ML, and Human-Computer Interaction (**HCI**) (Adomavicius. G, 2005). They share the same goal of helping the user to carry out tasks, discover products, infer preference models, etc. RSs are combinations of numerous algorithms that are used to gather different types of data from users (implicitly or explicitly). They predict the user's preferences and interests. In general, an RS connects users and products in three ways as represented in Figure 2.2. For example, (Burke, 2002; Resnick, 1997) have described RSs as software tools and techniques that deliver suggestions for items that a specific user likes. They provide suggestions to the users to help them in various decision-making processes (Ricci, 2015).

Figure 2.2: Three Common Ways to Connect Users and Products. (Haoxian, F, 2017)

Furthermore, RSs consist of three elements, as shown in Figure 2.3 and involve various techniques to provide recommendations to users based on many standards, such as data source, purchase history, user profile, implicit and explicit data. Both implicit and explicit feedback is used to generate the user model and improve the RS's functionally. Yet, the explicit feedback technique is difficult to implement because users regularly ignore an item's rating and direct questions that are sometimes required to define the model. The implicit feedback technique can solve the explicit method issues since it doesn't require the user's direct involvement to set the preferences. However, it is less accurate in comparison to the explicit feedback method (Isinkaye, 2015).

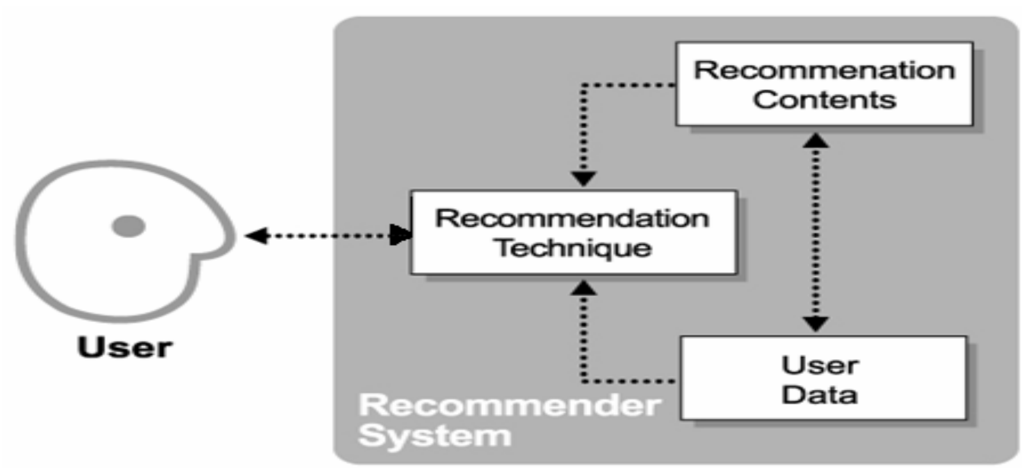


Figure 2.3: Constitution of Recommender System (Stuart Edward Middleton, 2003).

RSs can be classified by using three general factors, which are:

- Source of data used by SRSs, such as websites, social networks, or Web 2.0, and Web 3.0 like the global positioning system (**GPS**).
- Targeted data of both user and item, such as the user profile.
- Data extraction methods, which include explicit and implicit methods. To give recommendations in both methods, we need appropriate information about the user profile and his/her behavior (Manning, 2009).
    - Explicit Data Collection: Information is entered by users who are asked to rate an item. Users then choose one from two or more items and evaluate them on a differentiated scale. Once this is completed, they are asked to provide a list

of favourable items in a certain category. This method is more accurate since it was registered personally by the user.

- Implicit Data Collection: Uses system agents to capture the user's behaviors to extract their interests. It also observes online user purchases, behavior, history, number of views of an item as well as monitoring what a user is viewing in online stores, and the time spent on items. This method is not always valid, but the user is not required to input any information.

2.1.2 Filtering Techniques used in RSs

Many studies have developed filtering techniques, recommendation models, and methods to build social recommendation systems in different applications and domains. The main component of any proposed RS is its filtering technique. This usually filters structured information and organizes the data needed to get close to what the users are most likely to have suggested, as shown in Figure 2.4. The most used filtering approaches for RSs, in general, can be classified into three techniques, which are CF; content-based filtering (**CB**); and hybrid filtering, which combines two or more models to get more benefits from their features (Candillier, 2007). However, not all combinations of the basic models are available for the hybrid approaches. Some combinations can be redundant depending on the basic models (Burke, 2002). Filtering technique selection remains a challenge for hybrid models because choosing one mostly depends on the data type, situation, and background knowledge.

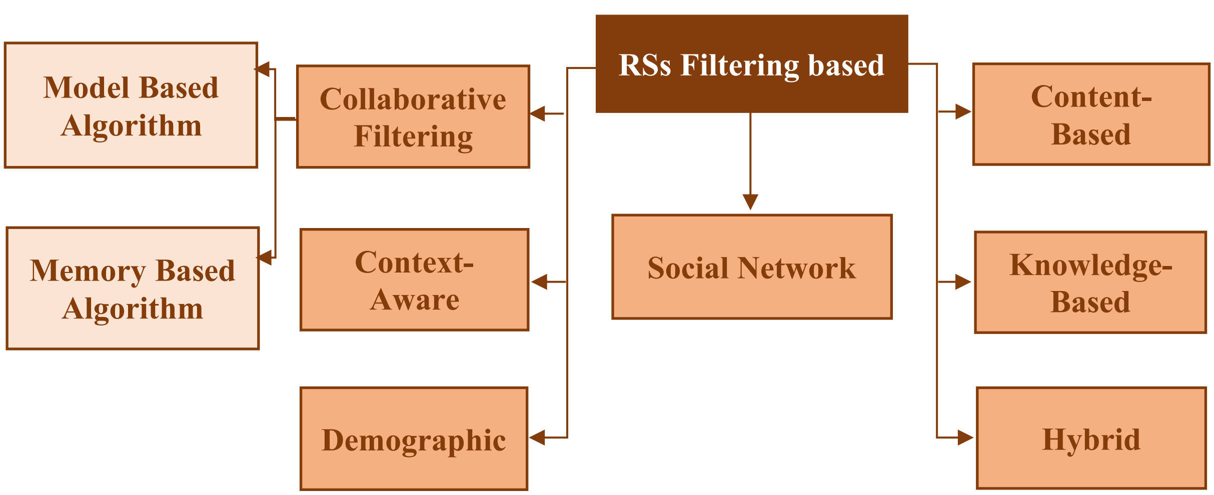


Figure 2.4: Recommendation System Filtering Techniques Based on Data Source.

CF is widely used in almost all RS types and is the most popular method, while CB filtering is widely used on textual items to extract relevant keywords used for future production items. Table 2.1 contains the fundamental description of the main RS filtering methods; for more details, refer to Appendix 1.

Table 2.1: An overview of basic recommendation system filtering methods.

Method	Definition
CF	-Based on users' past behavior and interest like rating, past purchases, opinions, and decisions made by other users to predict future interest towards a certain item or opinion. It basically tries to find the similarity between users' profiles and items. -Focus on algorithms that match items or people preferences. -Has two approaches which are user-based and item-based CF. -Mostly used in social networking sites like Facebook.
CB	-Uses keywords or other item's characteristics and features to recommend items and ideas with similar features. -Use features from the user's past liked items to infer new recommendations. -Focus on algorithms for learning user preferences and filtering a stream of new items. -Mostly used in e-Commerce websites.
Hybrid filtering	-Combine numerous RSs techniques to overcome the limitations of pure approaches and deliver better accuracy results. -Utilizes both the CF and CB filtering to predict the online behavior of the users in SM platforms in different ways such as combining both CF and CB features to optimize their strengths like in Netflix.
Demographic filtering	-Based on demographic filtering. -Tries to find the similarity between users' demographic information. -Categorize the user based on personal attributes and make recommendations based on demographic classes.
Knowledge filtering	-Recommend item based on specific domain knowledge, by mapping certain item features and users' needs and preferences like in Collaborative Advisory Travel System.

	-Has knowledge about how particular items meet particular user needs and employ this knowledge to recommend items based on inferences about each user's needs and preferences (Burke 1997; Burke 200; Burke 2007).
Context-Aware filtering	A special kind of knowledge-based system, when the context is involved as knowledge, is required for the recommendation. Include location-aware, temporal, trust aware, and other contexts.

2.2 Social Recommendation Systems SRSs

Today, it is easier for users to share their interests and opinions about many products and services that they use or purchase via social network sites or platforms. Furthermore, it is faster and more efficient for an individual to decide on various products. Incorporating RSs in OSNs has led to a specific kind of recommendation system called SRSs, which have been studied since 1997 and received great attention with the growth of SM (Kautz and Selman, 1997). SRSs have found their place in numerous applications, including online books, online purchases, research items, favorite restaurants, etc.

Various studies have defined SRSs differently. For example, (Giammarino, 2017) define them as systems that makes predictions about a person and/or a group of people based on their filtered information. Others like (Wang. S, 2017; Shu, 2018) define them as social relationship-based RSs. SRSs can examine users' activities on social platforms by analyzing comments and opinions about products and services or by using location, community trends, point of interests (**POI**), and friendship strength.

In their research, Peng (2019), proposed an SRS that provides real-time and accurate services (relevant users and interested items) based on a novel dynamic model. This DGE model can detect many features, such as temporal semantic effects, social relationships, and user behavior sequential patterns. This system developed a parallel incremental learning algorithm and an efficient query processing technique from a Threshold Algorithm to reduce the processing time of creating top-k recommendations from large-scale SM streams. During the system's evaluation, researchers conducted an extensive experiment through two real massive datasets and showed its efficiency compared to other state-of-the-art methods; however, they only considered local context information, like neighbors, in their process. There is other global information, like groups from which it can be affected.

Some studies focused on OSNs such as Google+ and Twitter, which provide the functionality of suggesting friends to people by utilizing their personalized information. Friend recommendations are an increasingly important component in studies of social networks, like in (Agarwal, 2013), who considered people's recommendations in the SM field. The work done by Lina (2017) presented an advertising RS that aimed to solve the issue of low efficiency in the era of big data, such as microblogging by offering a mixed recommendation algorithm with user clustering. They evaluated user information in a microblog called Sina, a short text platform to define the user's interest model and then recommend products according to their preferences.

They also applied the term frequency-inverse document frequency model (**TF-IDF**) and the Chinese word segmentation tool to divide the text into words without dealing with noise data that can easily affect the accuracy. As a result, their algorithm was efficient since it increased accuracy and recall rates in microblog advertisement promotion. None of the above advertising RSs use the dynamic context of the user's dialogue conversation since they recommend based on items the user finds most interesting, friend posts, feedback, and initial recommendations.

Figure 2.5: Sample Web Application (Sevli, 2017).

In addition, Sevli (2017) proposed an advertising RS to deliver content to Twitter platform users based on their dynamic data analysis. Basically, they tried to analyze tweets shared between a group of Turkish Twitter users and explored their profiles in real-time in order to recommend suitable content advertisements to them, as shown in Figure 2.5. They applied both the NLP and map-reduce method to define the user's interest area based on the generated content.

Researchers in this work extracted the most used word patterns, then applied a Heuristic Method as a content filtering method that grades the content of ads with weight values calculated based on the users' areas of interest and recurrence in the words they repeat with low repetition suitability. The map reduced method was used to define the word patterns in the user's shared content. It was then categorized to generate the user's area of interest. They developed an advertisement database that included particular keywords and related categories to recommend content ads based on these factors. This system was efficient; however, relying on extracting only the most used keywords list to define user's interests may lead to recommending useless advertisements that users may not be interested in at a specific time. Researchers were focusing only on Turkish Twitter users. Expanding this work is necessary to test its usability.

(Yuchen and Zhang, 2016) tried to enhance online advertisements by addressing the benefit of using social media marketing. They considered the directly targeted advertising channel to define more sophisticated social media marketing strategies by proposing a context-aware advertisement recommendation framework that integrated both the user's static personal interests and dynamic news feed (posted from the user or their friends), as shown in Figure 2.6.

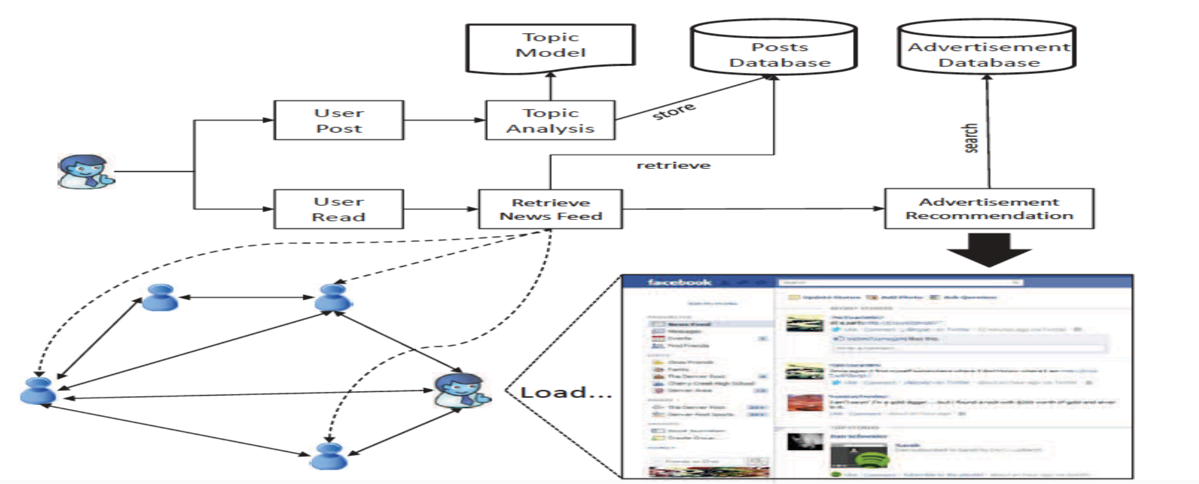


Figure 2.6: System Overview of Context-Aware Advertisement Recommendation (Yuchen, 2016).

They aimed to solve the problem of dynamically changing news feed content by designing three methods: the online retrieval algorithm, the top-k aggregation algorithm, and the safe region method. These are used to obtain personalized advertisement recommendations that match the user's preferences and meet their real-time requirements. They also conducted an extensive experiment during their system's evaluation over real-world social graphs with millions of users and billions of social links. These experiments reveal the effectiveness of the proposed solutions to support social media marketing strategies. Users may get the same set of top-k advertisements if the dynamic context of users stays the same or changes a little.

Furthermore, (Lad, 2016) proposed a novel diverse keyword extraction technique to find the most relevant keywords from the output of a real-time ASR system. It recommends relevant documents to participants during a short meeting. This method derives multiple, relevant, and diverse queries from an extracted keyword set search across English Wikipedia. The system extracted the most important keywords from the conversation and then applied the topic-based clustering method to build many topically separated queries, which ran individually, by using the ranking keywords, Figure 2.7 proposed the system architecture.

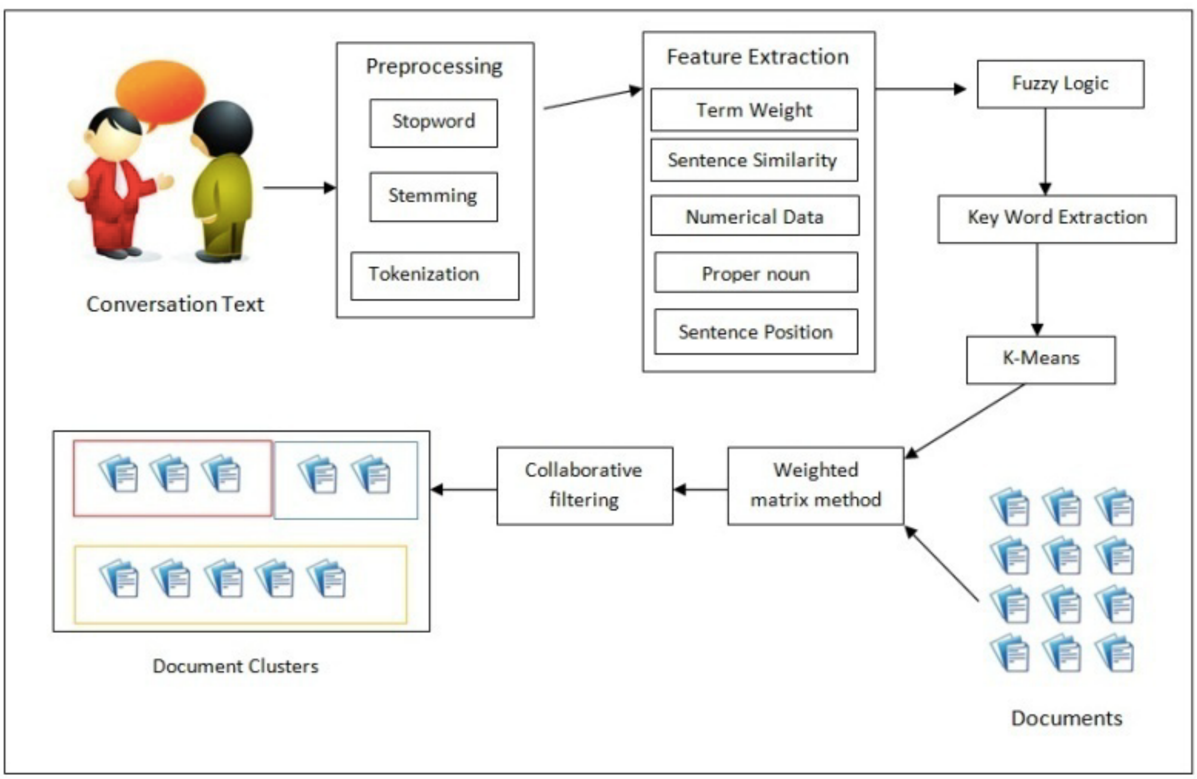


Figure 2.7: The Proposed Document System Architecture (Lad, 2016).

On the other hand, this system does not consider explicit queries in its recommendation process, which causes redundancy in its list of document results. Indeed, involving the SAR system to recommend documents can make errors when recommending items like documents to users; therefore, users wouldn't be satisfied by getting many recommended documents as a result of extracting a large number of topics from short fragment conversations. We believe that extracting fewer topics and considering the real conversation context can lead to better results.

Keerthana (2017) developed a document recommendation system from converted text from the ASR System that used both cosine similarity (word co-occurrence) and semantic methods, as well as the Latent Dirichlet Allocation (**LDA**) topic modeling that was implemented in the MALLET toolkit environment in order to extract the most significant terms form short conversation fragments. Initially, the topic model was used to determine the weights for the abstract topics in each conversation fragment. After extracting the keywords, the TM similarity methods were applied. In this work, researchers compared keywords extracted using different techniques, namely cosine similarity, word co-occurrence, and semantic distance techniques. They found out that word co-occurrence and semantic distance extract more relevant keywords than the cosine similarity technique.

During our research, we noticed that each previous study has its own way to recommend items for users. Some focused on offering several items without considering the user's personal interests while others worked to offer new algorithms to reach their final goal and enhance their implemented system. We plan to develop an SRS that can display items to the interested users at the right time, in an interesting and easy manner. Our proposed SRS should have some basic features, specifically personal, mobile, real-time, and text-based SRS. The following section introduces some rudimentary details about each feature, then an evaluation of previous studies that match most of our targeted features.

The following section has basic details about each included feature and the set of criteria that we take into consideration when reviewing and designing our proposed real-time SRS:

2.2.1 Personal based SRSs

Due to the rising need for proposing more personalized social recommendations, much research has been conducted in this field. There are many basic methods in personalized SRSs that researchers have used to ground in their systems such as those based on friendship strength in OSNs, those based on user's POI, those based on location and community trends, and others. Moreover, many social platforms today can offer increasingly personalized and context-sensitive recommendations to users (Khater and Gracanin, 2017). It is usually not an easy task, especially due to the massive amount of data needed, and keeping in mind the apparent limitations of noise-filtering, and the heterogeneous nature of information (Quijano and Sauer, 2017).

In OSNs domains like Facebook, the closeness of users can be measured before giving recommendations by exploiting the relationships between people in order to provide more personal and accurate recommendations, like in (Seo, 2017) friendship strength in this study is measured by exploiting three main features: contents generated by users, relationship information, and interaction information. In these methods, similarities and relationships between users are normally the primary areas of exploitation, measuring the level of relations and giving recommendations based on the level of closeness, or rather what they term as friendship strength.

Benefits

- Provide more personalized, relevant recommendations compared to other existing recommendation systems.
- Provide more context-sensitive recommendations.
- Some personalized recommender systems have incorporated the time-variant of the user's preferences in their model to get more relevant and accurate recommendations.
- Some personalized recommender systems used geolocation data in their model to solve the problem of cold-start and data sparsity.

Limitations

- The cold-start problem, because most of these systems use the CF techniques in their data collection and filtering.
- Data sparsity, usually there are few ratings and data to be analyzed by these systems.
- The problem of noise is sorting the relevant data from irrelevant data available on each person and then using relationships to give recommendations become very tricky.
- Sometimes end up giving unrelated recommendations in some regions/localities.

2.2.2 Real-time based SRSs

Real-time SRSs are found in a wide range of applications, not only in e-Commerce and OSNs. These kinds of recommendations include the time factor into their model before giving appropriate recommendations to users. They also analyze the needs of the user at that particular time (Akermi, 2016). For instance, (De Maio, 2018) proposed a time-aware collaborative filtering method adopted to evaluate what users might hold interest in when they browse through the Twitter platform. Text analysis services have been selected in this approach to annotate the content of tweets semantically and to track the notions based on the frequencies of being posted and forwarded along at the time. The time function is considered as an essential key in our system, this function important today to satisfy users by dynamically providing them with personalized recommendations, adding the time factor to the social recommendation can lead to more appropriate recommendations to the users by analyzing their needs at that particular time (Akermi, 2016).

However, many proposed SRSs didn't consider the current user's interests in their developments; basically in the text-based form recommendation. Sara (2016) Made a study to get the contextual recommendation customized through a time-awareness system. The system recommends various items to the user without waiting for initiated interaction from the user. Also, there are many existing works about real-time systems and some remarkable works are within mobile and web page contexts. The mobile framework proposes an approach for proactive context-sensitive recommendations that cover several domains and provide those recommendations when needed without the user having to initiate any query on their mobile devices.

Benefits

- Propose a dynamic recommendation for active users.

Limitations

- High velocity- the speed at which information flows is too high.
- Data sparsity- information is too much and very much unbalanced over different topics.
- Topic variety-sometimes distinguishing different topics is a challenge since the topics under discussion especially are too much and can lead to noise if not properly filtered.
- High sociality- users have too many social interactions and filtering.

2.2.3 Mobile based SRSs

Like in a computer environment, the need for RSs in a mobile setting is essential nowadays. Smartphones and tablets have become the major devices for users to access the Internet in their daily lives (Wang. D, 2012). In addition, users used mobile technologies in many aspects, such as communication, information acquisition, and entertainment, as they are directly connected with their family and friends. Users' widespread adoption of mobile technologies in several areas, for example (Law, 2018) stated that the hospitality and tourism field has the most mobile service usage in industry application. Mobile SRSs changed rapidly because of the growth of the internet, especially in the e-Commerce domain, by getting benefits of wireless networks and the development of mobile devices. However, there is a need to make it more suitable for the mobile device users and to solve the information overload problem, as well as the limited mobile terminal visual interface that significantly affects user's experience and satisfaction.

We should indicate that mobile social media (**MSM**) in the mobile market domain is moving swiftly due to its omnipresence and universal reach. Researchers considered the MSM as a part of the mobile marketing APPs which they describe as all mobile marketing applications that enable a user to create content (Yadav, 2015). In fact, the mobile business environment has changed and improved the marketing industry dramatically in recent times. For instance, various studies revealed that it derives the consumer's interests and spreads the company's products and advertisements. Several marketing enterprises are working on designing mobile applications, models, and services that aim to improve the mobile e-Commerce field in general, while also finding technical solutions, methods, security information, event management technology, and protecting consumers from cybercrime. Hence, using a mobile device for interaction purposes makes users feel more secure as they are directly connected with their family and friends (Dan, 2014).

Benefits

- Has different types of multimedia content. like different generations of mobile devices, 1G, 2G, 3G, and 4G.
- Can be used at any time and place. Like during events, conferences, and workshops.

Limitations

- Hardware and software development challenges, such as screen size and type, battery life, communications capabilities, and others.
- Has limited resources such as storage, which make it hard for users to search or manage large volumes of multimedia contents
- Can cause traffic accidents and may influence sensitive medical devices and cause serious health issues.
- The security issue can easily attack the Virtual Private Network through a vast number of networks interconnected through the line.

2.2.4 Conversational based SRSs

Conversational SRSs have become very popular, especially in this age of global explosion in SM and the social networking services where individuals and groups participate in the sharing of opinions and information. This system supports a natural interaction model, interactive dialogue, which is similar to human interaction. It implements its strategies depending on the user's goals. Conversational RSs have many interaction approaches in their requirements elicitation processes to identify recommendations. It also contains many approaches to display the best recommendation lists to users. Several SRSs adopted this approach; for example, the Entrée system uses it to recommend restaurants (Burke, 2002).

Moreover, the work conducted by Nguyen (2017) developed a method to extract information and then generate and provide appropriate recommendations to a particular group of people. This method also employs interactive and conversational approaches to gather information and use it to influence a group's decision-making process. Several adaptive approaches are employed in data gathering. The process of data gathering is similar to the above-mentioned methods, only that here, this information is used to generate recommendations, not for an individual, but a whole group and consequently, influence the group's decision-making process.

Furthermore, extracting valuable information is considered to be an important method that conversational SRSs use to suggest appropriate recommendations, for example, by extracting keywords that are related to a particular topic or item. Indeed, most used keyword extraction methods applied in conversational RSs are based on data mining, like the text-similarity method that aims to extract relevant data to be used in RSs. Text mining is the process of extracting appropriate information from huge sets of data. For example, (Nguyen, 2017) describes a method to obtain information and then generate proper recommendations to a particular group of people.

This method employs interactive and conversational approaches to gather information and use it to influence a group's decision-making process. Researchers (Lad, 2016) proposed a keyword extraction technique that finds keywords from a meeting's conversation transcripts and then recommends applicable documents to the participants. They applied the feature extraction method that aims to extract several topic queries to increase their system's ability to define at least one appropriate recommendation.

Nonetheless, the most mentioned work in conversational SRSs applied topic modeling and keyword extraction methods to define topics and recommend services to users. We tackled many algorithms and techniques that worked well but mainly detect the most important topics without considering if they show the right topic like in document recommendation, for example, those that consider most repeated or high-frequency words in their extraction process.

Benefits

- Alleviating the cold-start problem, conversational RSs gather relevant information much faster.
- Alleviates the limitation of data sparsity.

Limitations

The main limitation of conversational RSs is the complexity of the system itself. Even after the keyword extraction process from whole conversations, so many topics are generated from them. Then the system needs to sort out the topics into relevant recommendations and those that are irrelevant. Sometimes, the user just fined only one recommendation turns out as appropriate from a whole analyzed data.

2.3 Tackling the Cold-Start Problem in SRSs

With the massive increase in the use of SRSs, consumers will face several benefits and problems that influence their commercial activities through the use of OSNs, such as the cold-start problem. Researchers have tried to solve it by proposing many social network-based and SM analysis approaches such as merging CF with CB recommenders in hybrid systems (A. Schein, 2002), obtaining ratings from new users (A. M. Rashid, 2002), and using the social networks of users (S. Sedhain, 2014).

The cold-start problem occurs when the system has little information about the user, since the consumer hasn't rated any products yet, and when there is a lack of product ratings and client reviews (Rana and Jain, 2015). Several studies have proposed solving the cold-start problem by utilizing data from OSNs. The cold-start problem defines a situation in which the RS is unable to create meaningful recommendations to users, due to many reasons like inadequate information (age, gender, occupation), lack of demographic information, and lack of the initial ratings. It mainly occurs when a new user or item has just entered the system.

The cold-start problem is a critical issue to deal with in the e-commerce domain and several solutions have been suggested to solve it. Numerous works have addressed this issue for both user and item like in (B. Lika, 2013; Lucas, 2015; Revathy. V, 2019); indeed, research on the cold-start problem in RSs is a multidisciplinary field of research and includes Psychology, ML, and HCI. However, many of the suggested solutions have not focused on the problem of personalization and dynamic suggestions that we aim to offer in our proposed real-time SRS. Despite all the advances in many related areas, more work needs to be done to solve the user/item cold-start problem in many practical e-commerce platforms and mobile applications. In general, because of the insufficient user's information and interactions, the RS fails to comprehend and understand the user's interests and consequently gives poor quality or random recommendations.

There are two categories of cold-start problems (Abbasi et al, 2014):

- User cold-start problem: occurs when a user registers to the system and the system has no information about his personal information, history, previous interactions, and preferences. This makes it fail to recommend anything because it couldn't form a user profile to match significant items and like-minded users. The new user cold start problem exists in both CB and CF filtering approaches. To build the user profile and produce suitable results, there should be enough user feedback, which is generally the item's ratings (Lika, Kolomvatsos, 2014).
- Item cold-start problem**:** occurs when items are new to the system and no ratings are available for them. This also occurs when the items receive very little interaction and no data about them (purchase history, comment, and like), even if they have been available in

the item set for a long time (Lika, Kolomvatsos, 2014). Hence, the item's content information is used in giving an early recommendation for the users. Item cold start problem is more obvious in the collaborative filtering approach, specifically those relying on an item-to-item method for creating recommendations. Researchers tried to solve the new user cold-start problem by effectively selecting items to be rated by users to quickly get their preferences to develop the recommendation performance (P. Melville, 2010).

In general, SRSs tried to solve the cold-start problem in different ways like incorporating the user's networks such as the user's social connections including users who share similar demographic information (J. Huang, 2011), who are trustworthy, and real-life friends (H. Ma, 2011). Still, such information might be difficult to gain due to privacy issues. For instance, (Qian, 2019) developed a novel framework for user/item cold-start problem called Attribute Graph Neural Networks (**AGNN**), that can generate the preference embedding for a cold user/item by learning the preference embedding from the attribute distribution with an extended variational auto-encoder structure (**eVAE**).

They also proposed "a gated-GNN structure to aggregate the complicated node embeddings in the same neighborhood, which enables a leap in model capacity since it can assign different importance to each dimension of the node embedding (P.2)". The experiment was conducted on two real-world datasets (Movie-Lens dataset and Yelp data) and applied two well-knowing evaluation metrics the Root Mean Square Error (**RMSE**) and the mean absolute error (**MAE**) to evaluate the AGNN model. The results show that the AGNN model yields significant improvements and outperforms others in comparison to many baseline methods. During some of the experimental procedures, the AGNN model obtained fewer results compared to other involving methods after increasing the ratio of the cold-start nodes in the graph because they are depending more on the user-item interaction graph rather than the attribute graph like in the AGNN method.

Behrooz. (2020) proposed a state-aware POI recommendation system called Sage used in several location-based services to solve the cold start problem when there is no information about the user is available to create a recommendation such as historical check-ins and social graph. "The Sage system reformulates the problem of POI recommendation as recommending explainable look-alike groups (and their POIs) which are in line with user's intent. It also frames the task of POI recommendation as an exploratory process where users interact with the system and their interactions impact the way look-alike groups are picked out". The Sage system was effective at providing personalized POI recommendations as well as being able to build explainable look-alike groups without relying on friendship aspects or check-ins of the user.

Bayomi (2019) presented a novel RS called Cold-start Resistant and Extensible Recommender approach (**CoRE**) which was "developed as part of collaborative research with Ryanair, the world's most visited airline website, where they have very limited information about their users' behavior concerning hotel booking" (P.1). The CoRE algorithmic approach included various recommendation methods, which were collaborative filtering, contextual suggestion, and content-based recommendations that aim to deliver personalized and contextually appropriate hotel room recommendations on the Ryanair Rooms. This approach was tested with a real-world dataset and showed effective and accurate results in extreme cold-start situations where there are shortcomings in the user/item data. The performance of the CoRE is slightly enhanced when not using feature weighting that would be more reliable in the cold-start scenario.

In their research, Sadia Zeb (2020) proposed a new matrix factorization method that combines the impact of item similarity to better factorizing the sparse rating matrix for item cold-start problems by merging association rules and clustering techniques. Two common techniques are adopted in this method which was the matrix factorization and the similarity measure (finding the nearest neighbors). Researchers in this work compared their developed method with other four CF techniques that consider user/item biases (Wang. D, 2014). The experiment was applied on two real-world datasets and the results confirmed that the proposed method added significant enhancements comparative to other evaluation methods; however, this method only is based on defining the similarity of the users' ratings.

Researchers in (Revathy and Pillai, 2019) proposed architecture to solve the cold-start problem in the recommendation system. They integrated both social networks and contextual data to build user and item profiles based on the client's preferences. They also used the CF method which included the contextual information in its clustering technique as well as the sentiment analysis to classify a user's reviews. It also collects the rating and preferences toward an item. This work considered different resources to collect data; however, researchers have not yet tested their proposed architecture in real-life online social network platforms or Apps.

Furthermore, (Benito-Picazo, 2016) proposed a conversational RS that aimed to solve the cold start problem based on users' information flow in the dialogue that allows an incremental elicitation of the preferred item features. This system doesn't require any information about users before they start to receive recommendations, since it considers product characteristics. Thus, when a user is interested in a specific attribute, it reduces the dataset to a subset that has items with said features and information. It produces a set of attribute implications and recommends a suitable item to the user. These researchers have also applied the knowledge-based recommendation algorithm called closure algorithm in their calculation process. The system can stop the process whenever it defines an appropriate item to recommend to a satisfied user or it can continue searching while deleting all previous suggestions to reduce the number of implications. This system only considers the product characteristics as a foundation in its suggested recommendation, which is not sufficient to satisfy the user's current requirements.

In our work, we aim to suggest a possible solution to the user cold-start problem and provide more accurate recommendations. The cold-start problem is described as the difficulty in bootstrapping the RS for new users or new items. Furthermore, we aim to understand the real meaning of a given text, not just extract a list of related keywords. The following part has a brief description of NLP and the TM method.

2.4 Natural Language Processing NLP

NLP is a field that combines the power of computational linguistics, computer science, and artificial intelligence to enable machines to understand, analyze, and generate the meaning of natural human speech as it is spoken. It aims to extract beneficial information from a large amount of data without physical human involvement of any kind. The first actual application of the NLP technique took place in the 1950s in a translation from Russian to English that contained numerous literal transaction misunderstandings (Hutchins, 2004).

More recently, there have been massive developments in the NLP domain, including both the rule-based system and statistical NLP approaches, which are based on ML algorithms for text mining, information extraction, sentiment analysis, etc. Moreover, some typical NLP real-world applications currently in use include automatically summarizing documents, named entity recognition, topic extraction, relationship extraction, spam filters, topic modeling, and more (Farzindar and Inkpen, 2015). These applications use many algorithms such as support vector machines (**SVM**) (Jiang, 2009), Naive Bayes (Witten, 1999), and others.

Below is a brief description of some essential natural language level tasks:

- **Machine Translation**: Used to translate from a basic language text into another language.
- **Textual Entailment**: Presented by (Dagan, 2005) and defined as the relation between two natural language tasks such as word sense disambiguation or question answering.
- **Part-of-Speech Tagging (POS)**: Presents labels like nouns, verbs, adverbs, etc. that are allocated to every token found in documents. It is also called grammatical tagging or word-category disambiguation and has many existing systems such as VOLSUNGA (Derose, 1988).
- **Natural language understanding (NLU)**: It is the post-processing of text after using NLP algorithms and context from ASR, conversations, personalized profiles, and similar tasks to understand the meaning, and then send output responses. NLU is considered to be one of the most used types of NLP recently because it considers commercial interests in their different applications such as automated reasoning, text categorization, and content analysis of subjects. One example of this task is RASA-NLU, which is an open-source discourse/dialogue manager framework.
- **Information Extraction (IE):** Defined as the automatic extraction of information from unstructured sources. Information Extraction is more of a NLP and ML problem where you train the machine to extract hidden information from the raw text. This task is used to extract the relationships between entities and to extract entities that have the most popular forms (called named entities) like a person's name, location, and companies, as promoted in MUC (Chinchor, 1998). Hence, many other subtasks combine multiple sub-tasks of IE to achieve a wider goal. Some examples of open-source information extraction software are Ollie (Mausam, 2012), which extracts binary relationships from English sentences.
- **Sentiment Analysis:** Known as topic mining and used to determine the emotional characters in text such as sadness, happiness, fear, etc. This task is useful in SM monitoring to obtain more information about people's opinion on a specific subject. For instance, BrandWatch Analytics Monitoring Tool[1] and social media Sentiment Analysis Tool (**SMSAT**) that searches for a specific keyword in several social media websites

https://www.brandwatch.com/

and then analyzes the output to define people's opinion about this brand and determine whether it is positive or negative. Figure 2.8 shows the result of [Dr. Inkpen] as a keyword.

Figure 2.8: Social Media Sentiment Analysis Tool.

- **Text Summarization:** Used to summarize text from articles or documents to outline the most significant points and the general meaning to reduce the required reading time. Text summarization methods are classified into two classes: extractive and abstractive. Each one includes several methods such as cluster-based method, graph-based method, text summarization with fuzzy logic method, tree-based method, and query-based summarization in the extractive text summarization approach (Gambhir, 2017).
- **Text Segmentation:** Used to extract words, sentences, and topics from written text. An example of an open-source word segmentation tool is Kyoto Text Analysis Toolkit (**KyTea**) which can perform both word segmentation and tagging like POS tags in the Japanese and Chinese languages (Kytea, 2013).
- **Question Answering:** Computer systems built to answer human questions in a natural language such as Eagle Eye as shown in Figure 2.9 as an example

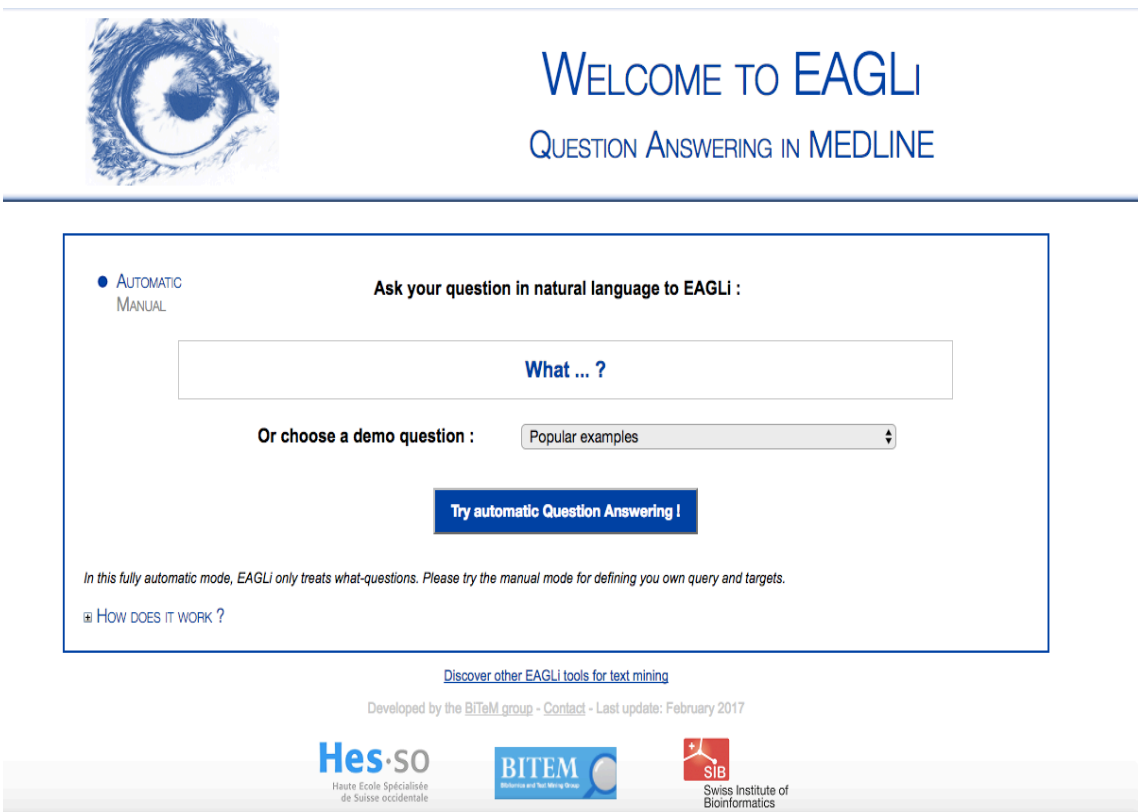


Figure 2.9: A snapshot of EAGLi question answering system.

2.4.1 Topic Modeling Method

(Miriam, 2012) Has described topic modeling as a method for finding groups of words (topics) in corpus text. There are several topic modeling algorithms used to discover the hidden thematic structure from a corpus of text such as latent semantic indexing (**LSI**) that is considered as the basis for the topic modeling development, but it is not a probabilistic model. There are two other main topic models; the first one is called Probabilistic Latent Semantic Analysis (**PLSA**) that was introduced in 1999 by Hofmann, for document modeling based on the LSI model (Hofmann, 2001). The second is the unsupervised probabilistic model called Latent Dirichlet Allocation, which was introduced by (David, 2003) to extend the PLSA model but was a complete probabilistic generative model. However, it is hard to use the PLSA model in new invisible documents because it does not provide any probabilistic model at the document level.

We reviewed the most common TM applications, tools, and algorithms as applied to OSNs. For example, (Liang, 2010) compared the performance of the LDA method and author–topic models on the Twitter platform. Jaffali (2020) presented a summary of social network data analysis, including its essential methods and applications in the context of structural social media data analysis. They divided the social network analysis methods into two types, namely structural analysis methods (studies the structure of the social network like friendships) and added-content methods (studies the content added by users). Likhitha (2019) presented a detailed survey covering the various TM techniques in SM text and summarized many applications, quantitative evaluations of various methods, and many datasets containing short content and documents which present various challenges. Furthermore, (Fiky H.E, 2021) reviewed many recommendation system challenges and techniques in the movie rating application. Also, (Kandukuri, M., 2021) provided an overview of textual data analysis and application that are related to the public relations campaign “Mann Ki Baat” in India.

Table 2.2 presented several related works in the SRS area. However, our work not only focuses on the review of TM tools, applications, and methods, but also includes several evaluations applying many techniques over short textual SM data to determine which method is the best for our future proposed system, which aims to detect real-time topics from online user-generated content.

Table 2.2: Shows some of the existing related works that revised the topic modeling method:

Related Work	TM Method	Evaluation method	Outcome
(Vrishal i. C, 2020)	LDA with Bag of Word (BoW)	Visual overview of extracted topics	-Aimed to discover the current trends, topics or patterns from research documents to overview different research trends. -The result shows that the LDA is an effective TM method for creating the context of a document collection.
(Santosh . K, 2019)	LSI LDA NMF	Perplexity Topic coherence	-Introduced TM methods and tools to the Hindi language. -Discussed many techniques and tools used for TM. -The coherence result of the NMF model was a little better than the LDA model. -The perplexity of the LDA model on the Hindi dataset is good compared to other evaluated TM methods.
(Aiting. X, 2019)	LDA	Perplexity	-Aimed to help Chinese movie creators to get the psychological needs of movie viewers and provide suggestions to improve the quality of Chinese movies. -Used the word cloud as a visual display of high-frequency keywords in a text which gives a basic understanding of the core ideas of text data. -LDA provides topics that deliver a good analysis of the Douban online review. -Used the perplexity method to determine the best number of extracted topics, set to 20 extracted topics.
(Dilip. S, 2020)	BoW TF-IDF Naive Bayes SVM Decision trees	Accuracy Precision Recall F-measures	-The Nu-SVC classifier outperforms all other included classifiers in the set of individual classifiers. -Random forest classifier outperforms all other included classifiers in the set of the case on ensemble classifiers. -The SVC classifier outperforms all other classifiers in the set of individual classifiers. -Random forest classifier outperforms the remaining ones.

	Nu-support vector classification		-Considered only two datasets, other datasets of different sizes need to be studied for better results.
(Robertus. N, 2020)	LDA NMF Task-driven NMF Plink-LDA The Non-Negative Matrix inter-joint Factorization	Purity Normalized mutual information (NMI) Pairwise F-measure	-It focuses on the review of the approaches and discusses the features that are exploited to deal with the extreme sparsity and dynamics of the OSNs environment. -Run the algorithms over both datasets 30 times and note the average value of each metric for comparison. -Most methods can achieve high purity value. -The NMF and NMijF having the best performance over the other methods AND NMijF performs the best results according to all the evaluation metrics. -Both LDA and NMF focus on the simple content exploitation of social media posts, main features (content, social interactions, and temporal).
(Yong. C, 2019)	LDA NMF Knowledge-guided NMF	PMI score and Human judgements	-Tested many numbers of topics (20, 40, 60, 80, and 100). - KGNMF model performs better than NMF and LDA.
(Aditya. A, 2019)	LDA NMF LSA	Precision, Recall, F-measure, Accuracy, Cohen's Kappa Score Matthews, and Correlation Coefficient	-Evaluated all topic modeling algorithms with both BoW and term frequency-inverse document frequency TF-IDF representations. -Used the Naïve Bayes classifier for the 20-newsgroups dataset and the random forest classifier for the BBC and PubMed datasets. -The results of the 20-newsgroups dataset LDA with Bag of Words outperforms the other topic algorithms. -The LDA model does not perform well with TF-IDF when compared to BoW. - The LDA takes a lot of time compared to LSA and NMF models.

2.5 Comparison to the Existing Studies

Many existing SRSs rely on the ML and NLP to define and recommend items that match users' preferences and to detect their interests in a real-time mode. They mainly identify the most important and/or most repeated keywords without considering whether or not they reflect the right topic. Indeed, several of the current methods focus on the text as sentences without considering the actual meaning of the sentence. We aim to define the practical context of the user's conversation by understanding the real requirements and needs during the conversation sessions. Accordingly, we decided to use the TM method in our work to extract the text features (topics in our situation) that can help to recommend more suitable and personalized items to users in real-time.

In SRSs, proposing the right preference items to a specific user at the appropriate time is a challenging subject. The predicted item will be dependent upon several processes and attributes such as the user's history, social relations, and others. SRSs can become more reliable if adapted capable interaction models and algorithms aim to analyze UGC and suggest suitable external tasks, for example, recommending advertisements to the intended user. However, users are not often satisfied with the initial recommendations because some of those models do not elicit the user's preferences and real requirements at the beginning of the interaction. There is still a need to study the UGC, such as conversation context in ONSs, to provide better-related tasks. We defined specific anticipated characters that are essential when designing an effective real-time SRS. Table 2.3 shows a comparison between our ChatWithRec system and other existing SRSs (2016-2021) that contained some of our essential regions.

Following are the main characteristic we considered:

- C1: Proposing a recommendation in a real-time manner.
- C2: Proposing recommendations from anywhere and at any time.
- C3: Use natural language understanding NLP/NLU techniques.
- C4: Support online UGC as a source of data.
- C5: Support keyword extraction technique as a recommendation acquirement.
- C6: Display recommendation that matches the user's attractiveness and personality.
- C7: Support conversational style.
- C8: Solving the user cold-start problem.

Table 2.3: SRSs principal characteristic comparison.

Related Works	Characteristic							
	C1	C2	C3	C4	C5	C6	C7	C8
(R.Preethi, 2021)	N	N	Y	Y	Y	N	N	N
(Peng, 2019)	Y	N	N	N	N	Y	N	Y
(Sadia Zeb, 2020)	N	N	Y	N	N	N	N	Y
(Volkovs, 2017)	Y	N	Y	Y	N	N	N	Y
(Fedelucio, 2018)	N	Y	Y	Y	Y	Y	Y	Y
(Pandit, 2016)	Y	N	N	Y	Y	N	Y	N
(Keerthana, 2017)	Y	N	Y	Y	Y	N	Y	N
(Nguyen, 2017)	Y	Y	N	Y	N	Y	Y	N
(Zihuan. W, 2018)	N	N	Y	N	Y	N	N	Y
(Sevli, 2017)	Y	N	N	Y	Y	Y	N	N
(Lina, 2017)	N	N	Y	Y	Y	Y	N	N
(Lad, 2016)	N	N	Y	Y	Y	N	Y	N
(De Maio, 2018)	N	N	Y	Y	N	Y	N	N
(Guo. B, 2016)	Y	Y	N	Y	N	Y	N	N
(Meng, 2018)	N	Y	Y	Y	Y	Y	N	Y
(Benito-Picazo, 2016)	N	N	N	Y	N	N	Y	Y
(Giammarino, 2017)	N	N	Y	Y	N	Y	N	N
(Xu, Xiaoying, 2018)	N	Y	N	Y	Y	Y	N	N
(Wu, Chao, 2017)	Y	Y	Y	Y	N	Y	N	N
(Behrooz, 2020)	Y	Y	Y	N	N	Y	N	Y
(ChatWithRec, 2021)	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**

The above table shows the need for more efficient methods and tools that can aid in detecting and analyzing the content of OSNs, particularly for those using UGC as a source of data in their developed systems. In addition, numerous studies in the SRS domain have tried to explain and solve the user/item cold-start problem which affected the recommendation system's effectiveness and performance and help in improving the relevance of recommendations by using several NLP algorithms and intelligent methods such as obtaining the user's history, tags, item's rating, social similarity, sentiment analysis method, user's activity, and preferences in order to predict their next recommendation. Indeed, many recent studies have employed online and offline UGC from OSNs to solve the problem. However, this issue is still unsolved as many recommendation systems predict users' preferences based mostly on their offline data and the user's preferences can be affected by numerous contextual features that should be considered when delivering a recommendation.

In our work, we aim to propose a real-time SRS architecture by merging CF with contextual CB recommenders in a hybrid method that good at predicting the online behavior of the users in SM platforms and provides more accurate recommendations. Hence, the time function is considered an essential key in our system. This function is important today to satisfy users by dynamically providing them with personalized recommendations. Adding the time factor to the social recommendation can lead to more appropriate recommendations to the users by analyzing their needs at that particular time (Akermi, 2016).

Table 2.4 presents the most similar, dissimilar, and latest SRSs that designed to enhance the commercial activities in the OSNs domain. Again, we believe that offering a real-time SRS that can understand the real context of a user's conversation dynamically is fundamental to defining and recommending interesting objects at the right time. Hence, in order to offer satisfactory services to the users, the SRS should effectively recommend new users/items when they enter the system for the first time.

Table 2.4: General Comparison of the most similar, dissimilar and latest works.

Related Works	C.1	C.2	C.3	C.4	C.5	C.6	C.7	C.8
Most Similar Systems								
(V.Revathy ,2019)	N	N	Y	Y	Y	Y	N	Y
(Fedelucio, 2018)	N	Y	Y	Y	Y	Y	Y	Y
Most Dissimilar Systems								
(Zihuan, 2018)	N	N	Y	N	Y	N	N	Y
Latest Systems								
(R.Preethi, 2021)	N	N	Y	Y	Y	N	N	N

Despite all the advances in related works, the interest in SRSs remains high. This research area needs more improvements, like the ability to deal with information overload and provide the user with more suitable and personalized recommendations in real-time. In different previous studies, users were often not satisfied with the initial recommendations because the users' preferences were not defined at the beginning of the interaction (Bridge, 2006). Some studies have proposed methods and algorithms such as common users preferences, similar items, and neighbors profiles that recommend items to users without considering their actual requirements and interests. Text-based SRS mostly relies on the ML and NLP such as topic modeling to define and recommend items that match users' preferences as well as to detect the user's interest in a real-time manner. Yet, they mainly identify the most important and/or most repeated keywords without considering whether or not they reflect the right topic. Each previous study mentioned in our work has its own way of recommending items for users without considering their actual personal interests.

2.6 Summary

SRSs have become an important area of research, as they can be used in many areas: within mobile apps or desktop applications, to enhance commercial activities, or to improve the user's satisfaction by dynamically providing them with real-time personalized recommendations, content, and services, in a remarkable manner. In this chapter, we discussed research on SRSs, as well as NLP methods that aim to predict the online topic from the user's OSNs, provide more accurate recommendations to satisfy online users, and suggest a possible solution for the user cold-start problem in SM platforms. The cold-start problem is described as the difficulty in bootstrapping the RS for new users or new items. Moreover, we reviewed TM, its common applications, tools, and algorithms that have been applied to OSNs. TM is defined as a type of statistical and unsupervised model that can quickly determine the latent topics within textual data. In addition, we reviewed other research on the real-time SRSs, including different recommendation techniques and how this thesis differs from the current research in that field.

Chapter 3 Thesis Methodology

The term methodology can be defined as “a system of principles, practices, and procedures applied to a specific branch of knowledge (Peffers et al. 2007. P.49)”. The proper methodology can aid researchers in fulfilling the requirements of a certain task or activity. Selecting the right research methodology is a primary task of the researcher because it is defined as a guideline to solve a particular issue using proper methods, tools, techniques, prototype development, and evaluation of the system. In our work, we are going to follow the Design Science Research Methodology (**DSR**).

Throughout this research, we performed a thorough literature review to identify the real-time SRS characteristics and solutions to derive the basic design components of our developed architecture. One of the methodologies that is appropriate to adopt in this research is the DSR (Peffers et al. 2007; Hevner et al. 2004; March and Smith 1995), which is generally used for performing research in the Information Systems (**IS**) field. The formalized DSR model is presented by Peffers et al (2007), which involves six activities structured in a nominally sequential order; however, researchers can essentially start at any step, and move to the next step. Thus, the nature of the research can help researchers identify their possible DSR possible entry points. The field of SRSs offers many opportunities for the DSR methodology to improve the effectiveness of recommendation items, services, and methods.

The following section presents the basic methodological approaches deployed through the work:

3.1 Literature Review Methodology

The literature review is a written summary of journal articles, documents, books, and other relevant sources that describes the past and present state of information on the topic of a certain study. It is the starting point and considered as the backbone of the research progress, because of this we add it within the methodology in this work to ensure that the included content is updated with the available development of state of the art as well as to aid in the decisions on follow-up work.

Chapter 2 and 5 included some related works about SRS and NLP areas, commenced in 2015, and has continued until the date of submission 2021, to ensure that the work is up to date.

We conducted an extensive background study to investigate the area of research that needs further exploring, by extracting the main gaps found in the literature and focusing on the contribution to this area. The next step after acquiring the relevant data is selecting the best tools, techniques to design and implement the proposed architecture of the SRS. Furthermore, we covered both SRSs and NLP domains. Indeed, the purpose of this study is to provide a better understanding of the trend of SRSs and to determine how NLP and topic modeling techniques benefit the commercial environment, The research methodology process shows in the following Figure 3.1. We focused on an article published between the years 2015 to 2021 since we aim to focus on recent research in this area. We considered searching for different library databases, which are:

- IEEE
- Web of Science
- ACM
- Scopus
- Science Direct
- Springer Link

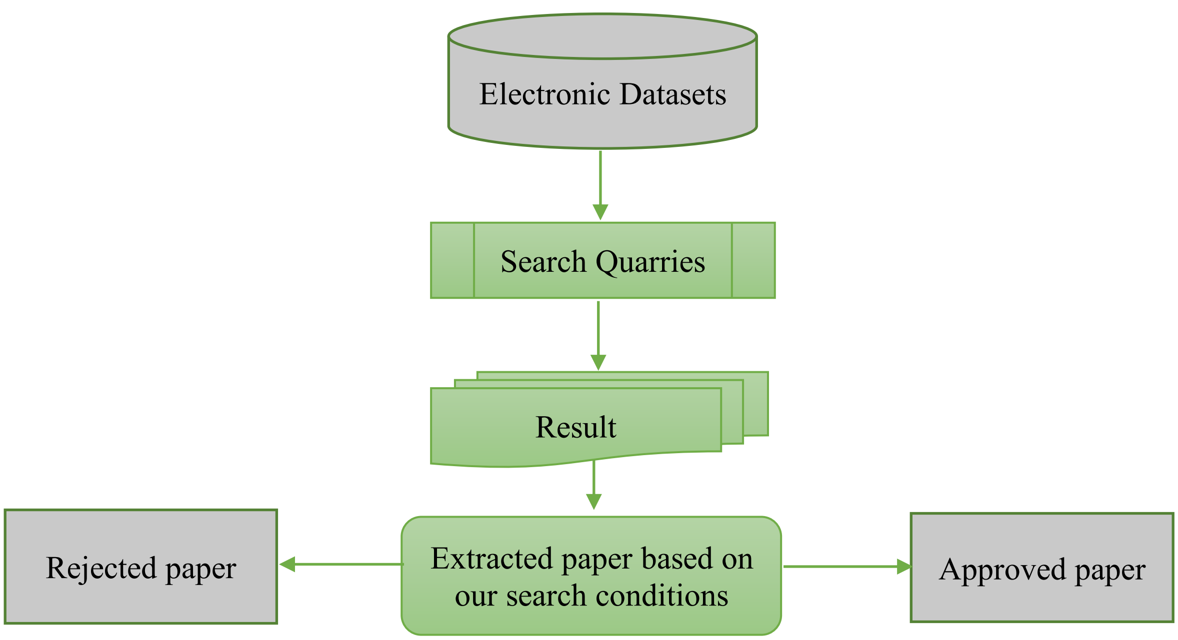


Figure 3.1: The Research Methodology Process.

The search was performed using the following keywords: “Social Recommendation System”, “recommendation system Architecture” AND “topic modeling”, “conversational recommendation system”, “classification & recommender system”, “cold-start” AND “RS”, “personalized recommendation”, “System Architecture”, “Advertisement recommendation system” AND “topic modeling methods”, “mobile recommendation system”, “the user cold-start problem”, “Contextual conversation analysis”, “cold-start problem”. We only consider conference papers and journals in the English language; other languages than English were excluded from this search. Moreover, we are going to follow the case-study methods in our future system evaluation, because it is mainly used for program evaluation to define systems capability and usefulness.

All the selected research papers were categorized into three separate groups, which are:

1) Articles containing research pertaining to real-time and/or text-based SRSs.
2) Articles with research pertaining to NLP/topic modeling subjects.
3) Articles containing research pertaining to both of the previous groups of research papers.

3.2 Design Science Research Methodology (DSR)

We followed the DSR as we aim to improve the commercial activities in OSNs, provide more personalized recommendation results, and enhance the user’s experience. This is done through the development of an artifact in the domain of the real-time SRSs. DSR is very applicable when it comes to creating IS artifacts that are innovative and solve real-world problems (Hevner & Chatterjee, 2015). DSR is a standard method that researchers use in the IS domain. It is about contributing new knowledge by creating advanced artifacts. DSR includes several steps that (Peffers, 2007) summarized, which are the problem definition and motivation step, the design and development step (create an artifact by defining its architecture and functionality), and finally, the evaluation step that measures how the proposed artifact can solve the defined problem.

According to March and Smith (1995), an artifact can be a concept, a method, a model, or an instantiation of them. “Constructs form the vocabulary of a domain, they constitute a conceptualization used to describe problems within the domain and to specify their solutions, a model is a set of propositions or statements expressing relationships among constructs. A method is a set of steps (an algorithm or guideline) used to perform a task. Methods are based on a set of underlying constructs (language) and a representation (model) of the solution space. An instantiation is the realization of an artifact in its environment." (March, S, 1995. P. 256).

3.3 Artifact Development

As we mentioned before, DSR is a problem-solving standard that aims to create new knowledge through the building and evaluation of information technology artifacts. The term artifact, as defined by (Gregor and Hevner, 2013), refers to any "thing that has, or can be transformed into, material existence as an artificially made object." (Gregor and Hevner, 2013. P. 341). In our work, we considered the SRS as the primary instantiation artifact that aims to understand the user's needs in a real-time manner and recommend appropriate recommendations to them. We started our search for an efficient instantiation artifact with an examination of the essential features that can enhance the proposed SRS, which are real-time, mobile-based, personalized, and text-based social RS, as described in Chapter 2.

Our objective is to design and implement an artifact architecture that can answer the research questions addressed in Chapter 1. This artifact is an SRS that uses the online UGC to provide more personal and accurate recommendations in a real-time manner as well as to solve the user cold-start problem. We followed the design and creation method because it is the most suitable way to answer our research questions. It was implemented through the design of an SRS artifact and proof of concept demonstration, besides involving diverse evaluation metrics experiments.

The design research field includes many proposed models and processes to operate. Hevner et al, define design research as follows "design science creates and evaluates IT artifacts intended to solve identified organizational problems (Hevner et al., 2004, P. 77)". Furthermore, Peffers (2007) "proposes a unified methodological guideline process that is a result of studying previous design research process models". By applying the model to four case studies the proposed process model is proven efficient and versatile. The process model itself consists of six different activities with four different entry points into the process depending on what is most applicable. Furthermore, "the fundamental outputs of the DSR methodology in IS domain are purposeful artifacts created to address important organizational problems. The artifacts must be described effectively, enabling the implementation and application of the artifacts in proper domains" (Peffers, 2007. P.22).

In this chapter, we followed the six activities of the methodology proposed by (Peffers et al. 2007) as shown in Figure 3.2, which includes a nominal process for creating an artifact, a real-time SRS for people in OSNs who are looking to receive related output such as product to their online generated content immediately, as well as contents to overcome the user cold-start problem and other issues in real-time SRSs. The DSR methodology can be summarized into several stages which are, (1) identify problem and motivation, (2) define objectives of a solution, (3) design and development of the artifact, (4) demonstration, (5) evaluation, and finally (6) communication.

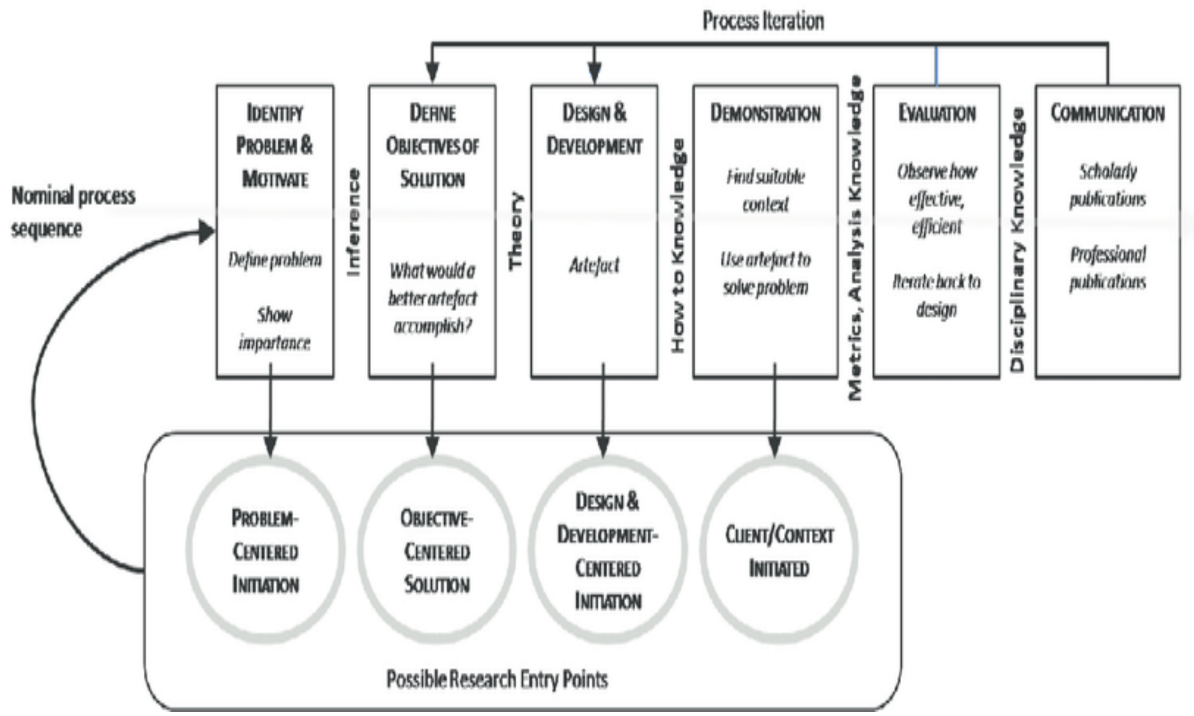


Figure 3.2: The Design & Creation unified methodological guideline process. Source: (Peffers and Vaezi, 2012. P.54).

We followed the DSR to create our architecture artifact as shown in Figure 3.3 as we aim to improve the recommendation activities in OSNs.

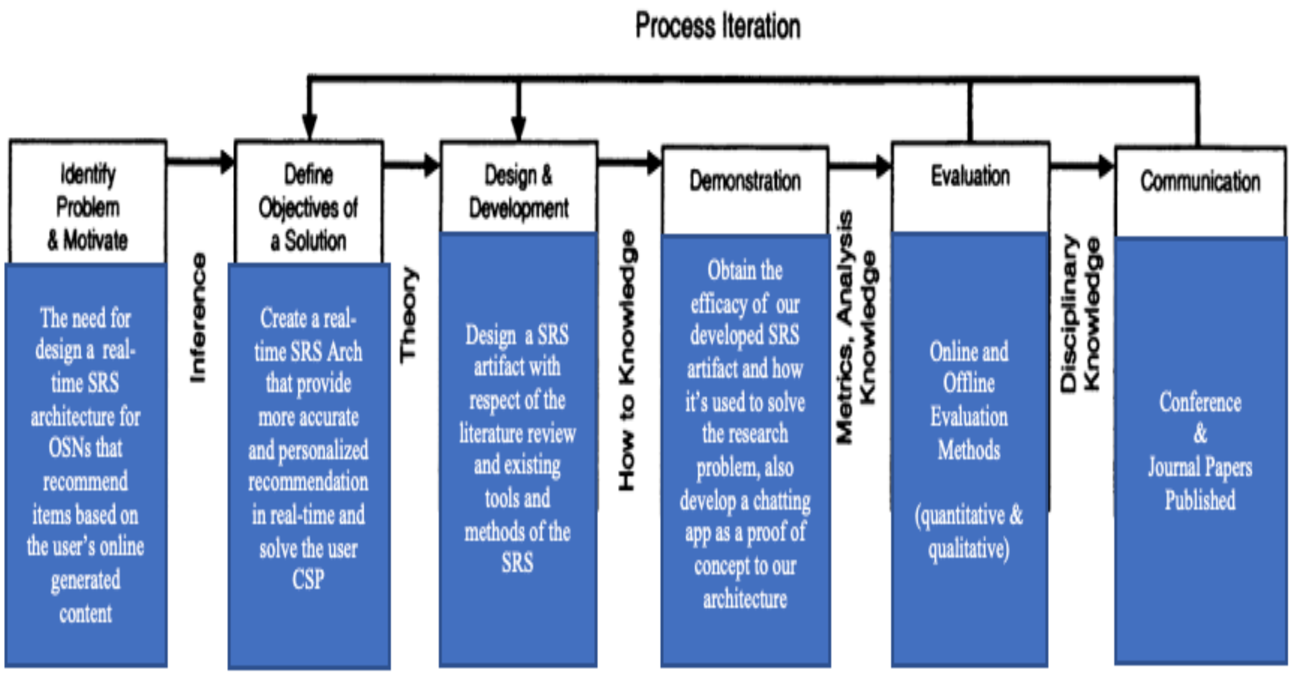


Figure 3.3: The ChatWithRec architecture artifact guideline process

In the following section, each activity is defined in detail besides we give a brief description of how we applied all activities in our research.

Activity 1: Identifying Problem and Motivation

Description:

"Problem identification and motivation. Define the particular research problem and justify the value of a solution. [...] Justifying the value of a solution accomplishes two things: Motivates the researcher and the audience of the research to pursue the solution and to accept the results and it helps to understand the reasoning associated with the researcher's understanding of the problem" (Peffers and Vaezi, 2012. P52).

Application:

SRSs have become an important part of improving the user's experience and revenue generation in recent times. The e-commerce markets that use OSNs platforms and applications provide many items that create a need for using RSs to provide suitable and more personalized recommendations to users. This increases the revenue generation for businesses as well as the user's satisfaction and experience. The most used approaches in e-commerce RSs are cookies and browsing history. For example, these provide related actions like recommendation ads in the user's second session. However, the growth in the e-commerce market creates a need for proposing the best possible SRS to ensure the best possible user experience and service for OSNs users.

In our work, the main problem is the need to design an architecture for a real-time social recommendation system that can obtain the user's current needs from their online generated-content in a real-time manner. The SRS is defined based on the observations of existing SRSs and the literature review which covered many SRS matters such as recommendation methods and NLP. We reviewed information about the SRSs area, its available methods, and techniques that we can use to obtain and utilize the user's information, preferences, and online generated-content.

We also will identify the critical SRSs aspects such as personalized, mobile, text-based, and real-time features. Hence, users are often not satisfied with the initial recommendations because some of those models do not elicit the user's preferences at the beginning of the interaction. There remains a need to study the conversation context to provide better-related tasks. Consequently, we defined several requirements that can be derived from the implementation of a sufficient artifact that is able to process online unstructured data in the OSN platform. We also consider the user cold-start problem at this phase and try to propose a possible solution by only considering the user's online generated content as a source of data in our recommendation procedure. Chapter 1 included an overview of our main motivations, objectives, research questions, and contributions.

Activity 2: Define Objectives of a Solution

Description:

"Define the objectives for a solution; infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible. The objectives can be quantitative, such as terms in which a desirable solution would be better than current ones, or qualitative, such as a description of how a new Artefact is expected to support solutions to problems not hitherto addressed. (Peffers and Vaezi, 2012 P55)."

Application:

Our second activity is creating an SRS that uses the user's online social media data in the recommendation procedure in a way that enhances the results of recommendation systems. We will studied many of the existing SRSs and their technologies to gain an understanding of the requirements for our architecture. Also, we will explain how our new artifact, which is a real-time SRS architecture is expected to enhance the social and commercial activates in OSNs, satisfy social media users, and provide a possible solution to the user cold-start problem. This is both a qualitative and quantitative objective. It is a qualitative objective since we are going to use online social media streaming data as an input source and we will be answering questions like how a specific user is linked to a specific recommendation such as ads. It is also a quantitative objective since we aim to improve and enhance the performance of real-time SRSs. Chapter 1 stated our SRS fundamental objectives, while Chapter 5 includes extensive details about our proposed architecture, including its components and other existing SRS architectures.

Activity 3: Design and Development

Description:

"Design and development. Create an Artefact. Such Artifacts are potentially constructs, models, methods, or instantiations. Conceptually, a design research Artefact can be any designed object in which a research contribution is embedded in the design. " (Peffers and Vaezi, 2012 P55).

Application:

This activity aims to fulfill the objective of Activity 2 because it is about the actual design and development of the artifact. This activity defines the artifact's required functionality besides its architecture to develop the actual object (Peffers et al., 2007). We specify and analyze the limitations and challenging issues in the existing literature to select and design the required features. Our artifact was designed with respect to current literature and existing development tools and services for social and mobile recommendation systems that aim to satisfy customer needs and increase the efficiency of existing services.

Furthermore, we identify different methods that are suitable for social recommendations, and set up the preliminaries for designing the artifact which intends to solve problems of major concern in the field of SRSs. The artifact is also designed as a conceptual architecture and a prototype is developed as a proof of concept of ChatWithRec SRS. Our design considerations, implementation process, and additional details are presented in Chapters 4 and 5.

Activity 4: Demonstration

Description:

"Demonstrate the use of the Artefact to solve one or more instances of the problem. "(Peffers and Vaezi, 2012 P55).

Application:

The demonstration aims to show the efficacy of using the developed artifact to solve problem such as case studies that enable readers to see how the artifact is being used to solve problems related to their research. Moreover, we develop a social network chatting app called ChatWithRec as a proof of concept of our architecture. It recommends advertisements only to subscribed users based on their generated content. Hence, this stage merged the recommender system with the dialogue platform to provide prototypes that represent many characteristics and features of each component of our proposed architecture. To validate our created artifact and define the usability of the designed architecture, we used different datasets to get proper results that related to the goal of Activity 2. Chapters 6 and 7 include more details about our included datasets.

Activity 5: Evaluation

Description:

"Evaluation; observe and measure how well the Artefact supports a solution to the problem. This activity involves comparing the objectives (so we need to define this in objectives) of a solution to actual observed results from the use of the Artefact in the demonstration. It requires knowledge of relevant metrics and analysis techniques. Depending on the nature of the problem venue and the Artefact, evaluation could take many forms. It could include items such as a comparison of the Artefacts functionality with the solution objectives from Activity 2, objective quantitative performance measures such as budgets or items produced, the results of satisfaction surveys, client feedback, or simulations. It could include quantifiable measures of system performance, such as response time or availability." (Peffers and Vaezi, 2012. P.56).

Application:

"This activity involves comparing the purpose of a solution to actual observed results from the use of the object in the demonstration" (Peffers et al., 2007). The evaluation stage is performed to determine if and how well the created artifact solves the proposed problem and to determine if the created artifact is complete, effective, and fulfills all system design requirements. To determine the performance of the presented artifact, we consider the process of the technical experiments that are considered as a common technique in the design system research (Peffers and Vaezi, 2012). Our evaluation includes two fundamental sectors (online and offline evaluations) to examine the performance, accuracy, and usability of the ChatWithRec system.

The first evaluation was to define the most useful NLP techniques to adopt in our developed architecture that can detect the right topic from the user's conversational session dynamically. We applied the offline testing which is usually utilized to enhance the model before online testing with live users. With offline testing, algorithms and models can be tweaked which may not be possible with a live system. Offline testing can also utilize a large selection of evaluation metrics and validation techniques to compute the quantitative merits of an approach.

Chapter 6 includes a brief explanation of many included topic modeling methods, which include the initial evaluation process, datasets, evaluation metrics, and its results. The following measured metrics were selected to evaluate the performance and functionality of topic modeling methods, recall, precision, and F-score. Our included methods are, LDA, NMF, PR, LSA, and PCA unsupervised topic modeling methods.

The second evaluation was to provide a possible solution to the user cold-start problem and to receive an appropriate advertisement recommendation during the user's conversation flow. It also is used to identify if the integrated method can fit the goal of determining the real user's requirements dynamically. We also applied the online evaluations in this evaluation process. Online testing evaluates the "look and feels" of a system. It has a basis in interface design as well as human-computer interaction and it also accounts for the qualitative examinations of a system. A case study evaluation is developed that recruits a group of users to demonstrate how well the performance of our developed architecture is. This case study operates as a usability scenario that shows our expectations from the developed prototypes of the proposed system.

It allows us to collect qualitative data and to define the user's within-subjects' interaction with the system to estimate the recommendation performance that was collected via surveys or questionnaires. It also includes many questions to measure the user's satisfaction and behavior toward the proposed system such as whether they like the system and have received interesting recommendations (Shani and Gunawardana, 2011). The following measured metrics are selected to evaluate the performance and functionality of the proposed real-time SRS, F-score, and Coherence score. Chapter 6 includes our final evaluation procedure, datasets, evaluation metrics, and its results.

Activity 6: Communication

Description:

"Communicate the problem and its importance, the Artefact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals, when appropriate. In scholarly research publications, researchers might use the structure of this process to structure the paper, just as the nominal structure of an empirical research process (problem definition, literature review, hypothesis development, data collection, analysis, results, discussion, and conclusion) is a common structure for empirical research papers. Communication requires knowledge of the disciplinary culture." (Peffers and Vaezi, 2012. P.56).

Application:

This is a final stage that includes communicating the importance of the solution, which helps to lay out the artifact effectiveness to be shared with peers and target users through conference and journal publications to receive feedback on the proposed designed architecture and compilation of the doctoral thesis. This thesis is the first and foremost method of communicating the knowledge identified in activity 6.

3.4 Summary

In this chapter, we presented our thesis methodology, described as a system of principles, practices, and procedures applied to a specific branch of knowledge (Peffers et al. 2007. P.49). We conducted an extensive background study to investigate the areas of SRSs that need further exploring, by extracting the main gaps found in the literature and focusing on contributing to this area. Also, we followed the DSR to design our SRSs architecture and develop an artifact that that is a proof of concept of that architecture. The proposed architecture aims to advance the user's social and commercial activities in OSNs, deliver more personalized recommendations to active users, and enhance the user's experience, which was done through the development of artifacts in the domain of the real-time SRSs. Finally, we followed the six activities of the methodology proposed by (Peffers et al. 2007) to outline our artifacts. Chapters 4 and 5 describe the architecture design and main technical implementation process of our system. All source code is available on GitHub[1] for further testing and extension.

---

https://github.com/users/Rania2016/projects/1

Chapter 4 System Architecture

In general, social recommendation systems aim to deliver possible items of interest to users in online social networks platforms. These recommendations are linked to the amount of available information about users and products such as the user's profile, purchasing history, social relationship, item rating, and other features. According to Gomez (2016), the Netflix company estimates its recommendation systems save up to 1billion dollars per year. The authors also state that "When produced and used correctly, recommendations lead to meaningful increases in overall engagement with the product (e.g., streaming hours) and lower subscription cancellation rates" (Gomez, 2016. P.7). In general, recommendation systems are considered a class of software systems that can offer meaningful recommendations for users and empower them with decision support (Khan et al. 2020. P.4).

In this chapter, we present the architecture of our proposed system and its underlying layers that represent a higher-level view that works as a third-party system to enhance the performance of the current social and commercial activities in OSN platforms and the social recommendations in general. We aim to find out if the user's social context can be used as an input source to offer more personalized items and overcome the issue of the user cold-start problem, considered a common drawback in SRSs, by integrating the social network platform, contextual data, NLP algorithms, and a recommendation engine. This section also presents fundamental design consideration of our proposed architecture besides the implementation details of a social mobile application that was developed as a proof of concept of our suggested system architecture.

4.1 Overview of the proposed SRS Architecture

SRSs assist users in finding attractive and suitable options for a wide range of preferences depending on the available information such as the demographic profile information. Nevertheless, many existing systems ignore the fact that user preferences can change over time according to his/her context and need. As well, some developed approaches were static and could not be directly extended to further recommended systems. In today's market, designing an efficient and useful

SRS is a challenging task since it includes highly dynamic data and may involve a large economic investment and time. (Rantanen, P. 2017) defined the architecture as “a collection of well-defined Application Programming Interfaces (**APIs**) contracts that specify the input and output parameters and methods required for communication as well as the structures and types utilized to represent the data”. A proper designing architecture should cover all system component communication, such as between the recommendation engine and the social network platforms and between components within the system.

Moreover, numerous extensive works have been done to develop new algorithms and methods that aim to provide more suitable and accurate recommendations for users, while other research has focused on the system’s architecture that supporting these algorithms exists. We reviewed many SRS architectures and considered some common and fundamental design processes in their implementations like prediction models and an architectural component that reflects the structure of the entire RS. For instance, (S. Sheth, 2010) summarized the common architecture implementation factors which were: a watcher (observing user activity), a learner (predicting relevant content), and an advisor (provides recommendations to users).

De Pessemier et al. (2014) “described a framework to detect the current context and user activity by analyzing data retrieved from various sensors available on mobile devices” (P. 7). On top of this framework, a RS was built to offer users personalized content consisting of relevant information such as points of interest, train schedules, and tourist information”. Maritza Bustos (2015) researcher proposed a generic architecture for developing educational RSs that aims to provide accurate information to students based on their preferences, user profiles, and learning objectives. This architecture is composed of many fundamental layers, including the presentation layer, the data access layer, the service layer, and the data layer, which contains many SM sources like, YouTube, and LinkedIn.

Unlike the SRS architectures described above and others mentioned in our literature review in Chapter 2, we aim to design an efficient and novel real-time social recommendation architecture that can work as a third-party system which can integrate several OSN platforms to provide suitable recommendations after analyzing the user short generated-content in a real-time manner. From a technological point of view, the development of such architecture is challenging since it combines many important aspects like social networks platforms, predictive analysis of SM content, and a recommendation system engine, which together provide better and dynamic recommendations to the target user in terms of performance compared to traditional SRSs.

In addition, other aspects are taken into consideration while proposing this new architecture, such as managing great volumes of data obtained from the OSN platforms dynamically and serving multiple actions (requests) within a short response time. Similarly, we studied other numerous design and development considerations that researchers consider to develop SRSs, such as web/app technologies, filtering mechanisms, service functions, and others based on the proposed system objective. More details will be explained later in this chapter.

We should mention that our system only analyses the online generated-content from subscribed users who aim to get benefits from their textual data. Furthermore, the extracted user conversation will not store in our database, which enhances the system's privacy and make the user more comfortable providing their generated content. Hence, only the user's preferences and device ID will be stored in the user's profile database for further analysis. Many existing recommendation architectures store the user content to provide more accurate recommendations in the user's second interaction which is hard to deliver real-time suggestions to the interested user.

During our practical review of many existing SRS architectures, we highlighted some of the main technical difficulties that we considered and tried to overcome while proposing this new architecture for the real-time SRS. These difficulties are:

- Using offline data mostly for developing SRS architecture. For example, using the user's past histories, known preferences, demographic characteristics, rating, or social relationships to recommend items.
- Lack of autonomous data update in the system operation.
- The slow capture of the personal interests of a user led to the repetitious recommendations.
- Lack of real-time reconfiguration and ability to support the user's customization.
- Lack of providing more suitable and personalized recommendation-filtering systems
- The fact that a recommendation seeker may ask for a recommendation.
- The fact that the recommendation system may create and push recommendations without the user's permission.

- Lack of an effective mobile-based SRS that users can rely on as an intelligent mobile/server.

Taking all the above technical points into consideration, it was decided that the method to design our ChatWithRec SRS architecture would have two main layers, namely a front-end computing layer (mobile/client-side) and a back-end computing layer (cloud/server-side); both are employed separately. This approach is certainly useful and more efficient because it provides simple, straightforward, and light app software that can handle many simple functions, as well as a powerful server that can handle the primary system's functions.

The front-end mobile computing layer includes two distinct roles, which are identifying the user's context (i.e., location and preferences) and providing an interface to display recommendations. Meanwhile, the back-end cloud computing layer provides various functionalities to the user, such as personalization, as well as storage and computation resources to produce recommendations that are communicated to the user online generated content in the mobile app. More details will be explained later.

The following aspects are taken into consideration while proposing the ChatWithRec architecture, which has original characteristics compared to other recommenders to help users to find attractive, essential, and appropriate recommendation options for a wide range of options:

- The system should support the user's online generated content as an input source.
- The system should gather user data and analyze them dynamically.
- The system should provide the recommendations within the application; this way a recommendation can be viewed and tested with minimal switching cost.
- The system should generate N number of recommendations based on the user's online generated content.
- The system should always return some recommendations at the same time as the conversational session.
- The system should handle much content (requests) per second at the conversational session times, then present the most plausible match recommendations.

- The system should avoid forcing the user to respond to the recommendation before continuing to work, since this type of system could be frustrating.
- The system application can be customized based on the user's preferences, besides exposing a simple and easy-to-use online and mobile user interface.
- The system can adopt many TM algorithms.
- The system can support suitable types of mobile ads with predefined ad templates and advertisement spaces.
- The system application does not take much time to detect topics then provide recommendations that are useful to the user.
- The user should be able to act on the recommendations when it is convenient for them.
- The system can be tested using common cross-validation methods.
- The system should consider the user's interests and preferences.

In general, three main components used to design a SRS architecture which is: (1) input (where user behavior and preferences are acquired, explicitly or implicitly); (2) process (where recommendations are made); and (3) output (where recommendations are offered to the user). In our work, the user interacts with the client application through the interface. The interface provides the user with the opportunity to chat, receive recommendations, and interact with it by opening it or closing it during his conversation session. The client application provides a list of recommendations that are related to the user's chat content. The server application needs to produce recommendations based on content received from the client application and the user's preference; this can be done by using a hybrid RS.

Based on the above system considerations, we proposed the general architecture of ChatWithRec SRS that classified into five fundamental components as shown in Figure 4.1. The system has several layers, namely:

❖ The Online Social Networks Layer

- ❖ The ChatWithRec App Layer.
- ❖ The Personal Assistant Server Layer has three basic modules which are:
    - Conversational Analysis Module.
    - Advertisement Recommendation Module.
    - The User Profile Module.

Furthermore, we are going to include two conversational models in our proposed ChatWithRec SRS which are:

- The Pure recommendation model.
- The Hybrid recommendation model. (Contain both user's content, location, and topic's preference as additional components to filter more the user's final recommendation).

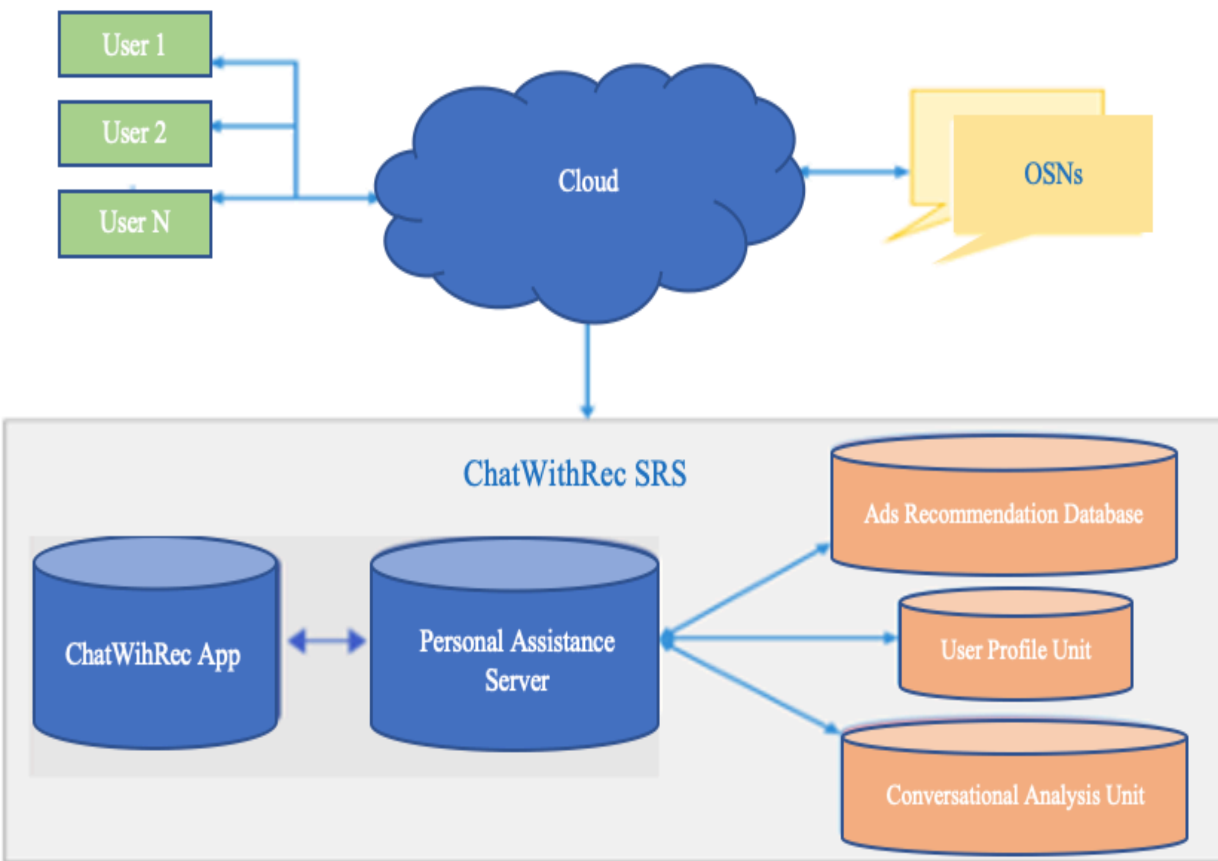


Figure 4.1: ChatWithRec general proposed architecture.

4.2 General Overview of the ChatWithRec SRSs Architecture

4.2.1 The Online Social Networks Layer

OSNs give users the ability to communicate with each other, make posts in the form of texts, images, videos, share thoughts and trends, etc. OSNs usually employ many themes in their recommendations; for example, location, interests, and occupation. ONSs can be categorized as social services, such as ratings, group buying, and social messaging services. The most popular OSNs that are used daily are LinkedIn, Pinterest, Instagram, and Snapchat, as well as Facebook, which is considered to be the most popular personal social network website and the most commonly used social messaging service.

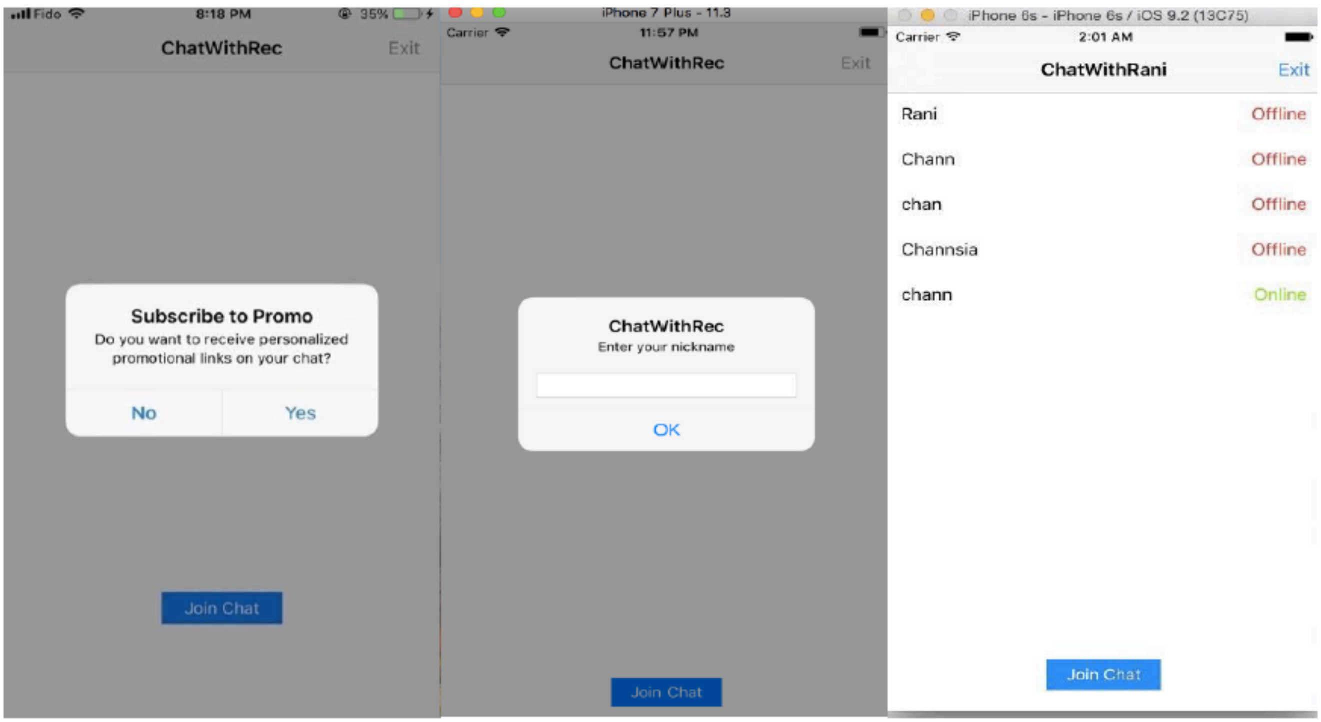
UGC can improve the prediction performance in SRSs. The analysis of the users' posts on their profile has become an important topic (e.g., detecting bad comments and recommending relevant products); however, most existing research concentrates on classifying long text data. Moreover, OSNs in general have numerous limitations including the cold-start problem, privacy, security issues, and data sparseness, among others.

We aim to integrate the users' social network applications and analyze their text-based interactions to suggest related content. Several researchers have addressed the cold-start problem for both users and items, considering it to be a long-standing problem in social recommendation systems. These researchers have taken several approaches to overcome this drawback. Some of the approaches include users' social connections, profiles, and demographic information. The cold-start problem occurs when the system has no information about the user, since the user hasn't rated any products yet, and when there is a lack of product ratings and user reviews.

Several studies have proposed solving the cold-start problem by utilizing data from OSNs; however, it is difficult to define and predict users' preferences from their data. We developed our social chat app called ChatWithRec as a conceptual solution to the cold-start problem, which is the main focus of this work. Our future goal will be to integrate other OSNs for a specific user who subscribes to our server to make more personal and accurate recommendations dynamically.

4.2.2 The ChatWithRec App Layer

This layer handles the collection of the required contextual data and interacts with the user; this also includes delivering the recommendation results. We developed ChatWithRec SRS, a mobile chat app, as shown in Figure 4.2, which is compatible with the iOS mobile operating system and built using Xcode-10.2 and Swift-5. The application mainly includes two functions, which are gathering online generated data from the user and acting as a personal assistant. ChatWithRec functionalities include collecting the user's conversations, then sending them to the personal assistance server while displaying results from the server to the user.



(a) (b) (c)

Figure 4.2: A screenshot of the app interface and user's list.

After installing the ChatWithRec app, a notification message will be shown to the user, requesting their agreement to receive relevant content (advertisements in this case). Users of the app benefit from having their conversations analyzed to detect topics and receive conversation-related content as they continue chatting. Through our app, a notification message will be shown directly to the user requesting their agreement to receive relevant tasks based on their conversation analysis. We should mention that through the sitting, the user can activate/deactivate this feature. Later, users can simply login by creating a username and then starting conversations with any active user who has also installed the ChatWithRec application.

Through this layer, users can interact with each other and receive suggested recommendations during their conversation sessions. In addition, the advertisements can be displayed in real-time after the system explores the keyword contexts to define a suitable advert for a specific interested user. The Node.js is used for this layer. This open-source JavaScript that allows JavaScript to run on a server, the Node Server is hosted on DigitalOcean where it is running on top of PM2. PM2 is a production process manager for Node.js applications as the server-side environment with a built-in load balancer. It also keeps applications alive forever so there is no need to reload them without downtime. “Node application rely on big processes with a lot of shared states and offers developers a powerful way to develop networking applications that will perform well compared to mainstream solutions” (Rauch, 2012). Figure 4.3 shows the process the user follows to start a conversation with others in our ChatWithRec chat app.

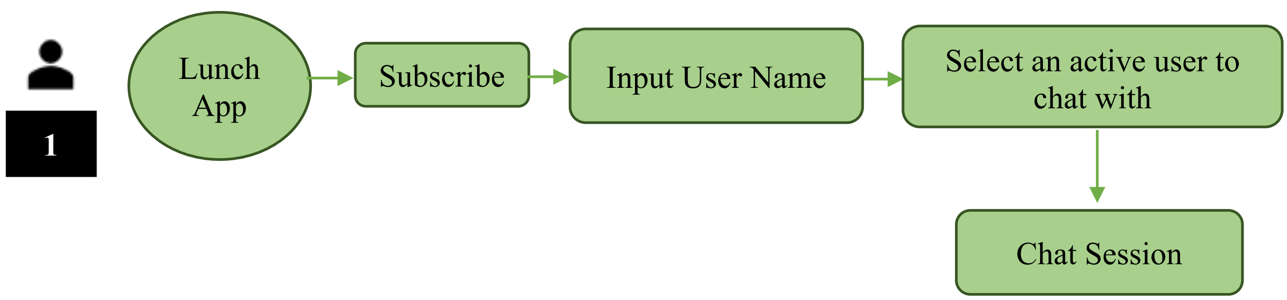


Figure 4.3: The process user follows to install and start ChatWithRec App.

4.2.3 The Personal Assistant Server Layer

This layer constitutes the heart of ChatWithRec system, which is developed to monitor the user’s conversation in real-time and determines suitable output recommendations to the connected user based on his textual generated content and preferences. Hence, this layer is considers a decision module in our proposed system, as it is responsible for providing the appropriate mappings between all components. In this layer, the pipeline of APIs is used to create separate corpora of words for both a user’s generated content and for each advertisement recommendation.

Based on the extracted topic of the user's conversation session, our system filters the most appropriate recommendations and sends them to the user. This occurs after showing a pop-up notification message to only interested users in their chat window, asking if they are interested in seeing related suggestions like advertisements. If the user selects yes, a suitable banner ad will be displayed. Otherwise, the user will continue chatting without receiving any ads. Users can access the advertisement page by clicking on the banner advertisement message while also being able to modify or decline to receive specific ads.

4.2.3.1 Conversational Analysis Module

This module is written in Python. Python is one of the most commonly used programming languages today and many organizations like Google, Facebook, and Amazon make use of it as it has a clean structure, a flexible design, and a wide library. For conducting the user textual analysis, we applied Scikit-learn, a free software ML library for Python languages. We should mention that most of our reviewed works applied the LDA model in Python and Java environments and not in Xcode environment. To overcome this challenge, we implemented the LDA model in Python and then integrated it into our Xcode server environment. It is available online in the ChatWithRec SRS GitHub account.

In addition, we applied different topic modeling algorithms in our proposed ChatWithRec SRS and compared our result with a collection of current topic extraction tools, as shown later in Chapter 6. In the beginning, we selected a short and straightforward conversation dialogue from Facebook OSNs (Zuckerberg, 2004). We should mention that each tool has a different process to filter the keywords depending on the corpus, document length or categories. For example, FiveFilters.org (Chen, 2005) used word occurrence and word count in its final output. Also, some extraction of open-source software tools required to clean up the text (tokenize the text) by removing capitalization, punctuation, numbers, and stop-words such as “a”, “the”, “and”, etc. We experimented with many existing keywords and TM tools to determine the most appropriate tool to integrate it into the ChatWithRec SRS.

We integrated the Gensim topic modeling toolkit in our ChatWithRec SRS based on our previous investigation. Gensim is defined as an open-source vector space and topic modeling toolkit which is implemented in Python and aims to extract semantic topics from text-based documents. It has been used in many recent studies because it operates faster than other topic modeling tools like Mallet and includes several kinds of unsupervised models that were implemented in our server such as LDA. Figure 4.4 has the key procedure of the conversational analysis module.

We also applied LDA because it is the most widely used model used by search engines and is considered the most popular topic modeling algorithm in real-life applications. In addition, the LDA model can provide accurate results, be trained online, and be incorporated with many assumptions such as the Bag-of-Words (BoW) and TF-IDF.

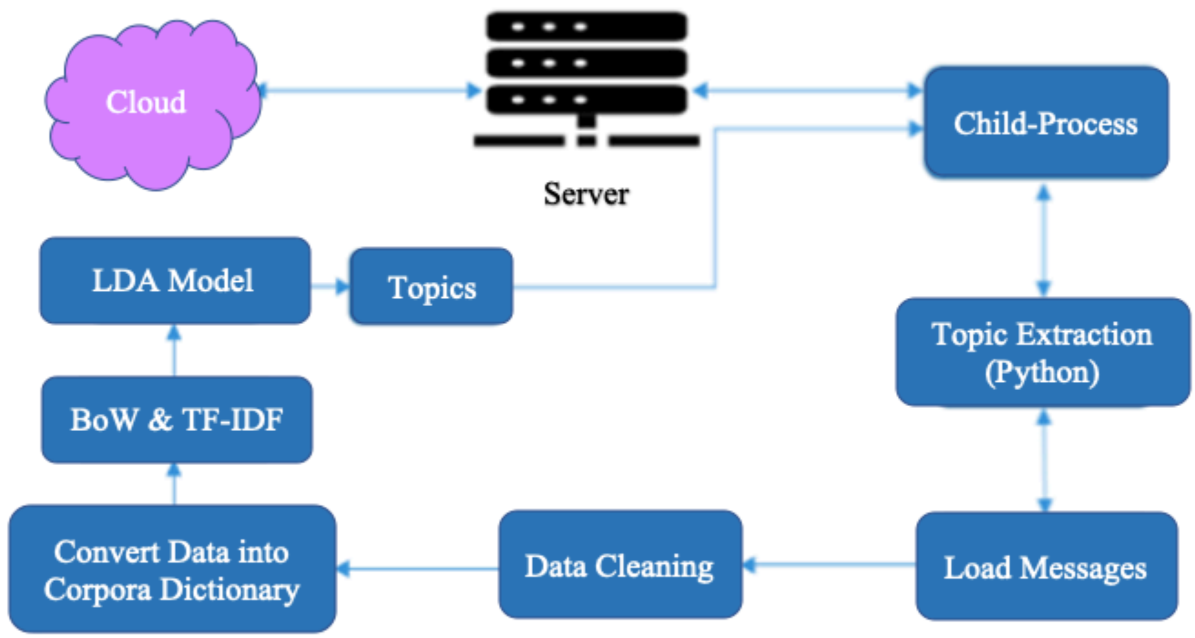


Figure 4.4: Main process of the conversational analysis module.

4.2.3.2 The User Profile Module

This module automatically builds the user profile based on his/her preferences. In this unit, we collect users' feedback to capture their preferences about different kinds of advertisement recommendation through the user interface. In general, a user profile works as an additional information filtering method in our recommendation procedure and it allows mapping the user to one or numerous advertisement categories and sites in a fixable and easy way. Users can customize their delivered task-related actions, advertisement in our case, based on their preferences and interests.

There are diverse types of user information techniques (explicit or implicit) that are used to determine users' preferences and interests. For example, the systems can gather user interests or preferences from user feedback (e.g., sets of keywords, user rating data, demographic information, topics of interest, questionnaires, and interviews) (H. Liang, 2010).

In our designed architecture, we used the explicit personalization approach because it is widely used to profile users' preferences and can reflect the actual user needs (G. Jawaheer, 2014). The subscribed user can select the list of advertisement categories and sites he/she wants to receive based on exploring his conversational session. We also used the keywords method, which is the most common representation for user profiles. Hence, each keyword represents a topic (ads category) of interest to the user followed by a site of popular ads websites.

This module involved the following steps:

- A user creates his/her individual account by entering a user-name when downloading the ChatWithRec app.
- After subscription to the system, the user explicitly receives feedback for each of the advertisement categories and the site is collected.
- The users submits the results to the server for further analysis.
- Builds the user's profile based on their feedback.

4.2.3.3 Advertisement Recommendation Dataset Module

Research by (Yera, 2017) highlights the current need to improve online advertisements, based on statistical studies that show a decreasing rate regarding the click-through rate of banner online advertisements, from 3% to 1%. In the same way, other researchers (Shu, 2018) and (Agarwal, 2013) indicate that having minimal cognitive capabilities to process online advertisements can lead to less attention from users. The main idea behind our ChatWithRec SRS is to match a detected topic from a user's conversation with a relevant ad that will be presented to him/her without interrupting his/her conversation flow. Topics extracted from the conversational analysis module are passed to the chat app using socket.io, a JavaScript networking library that runs the server side that is used to build our ChatWithRec app and to request an advertisement. Figure 4.5 shows our diagram for suggesting a recommendation.

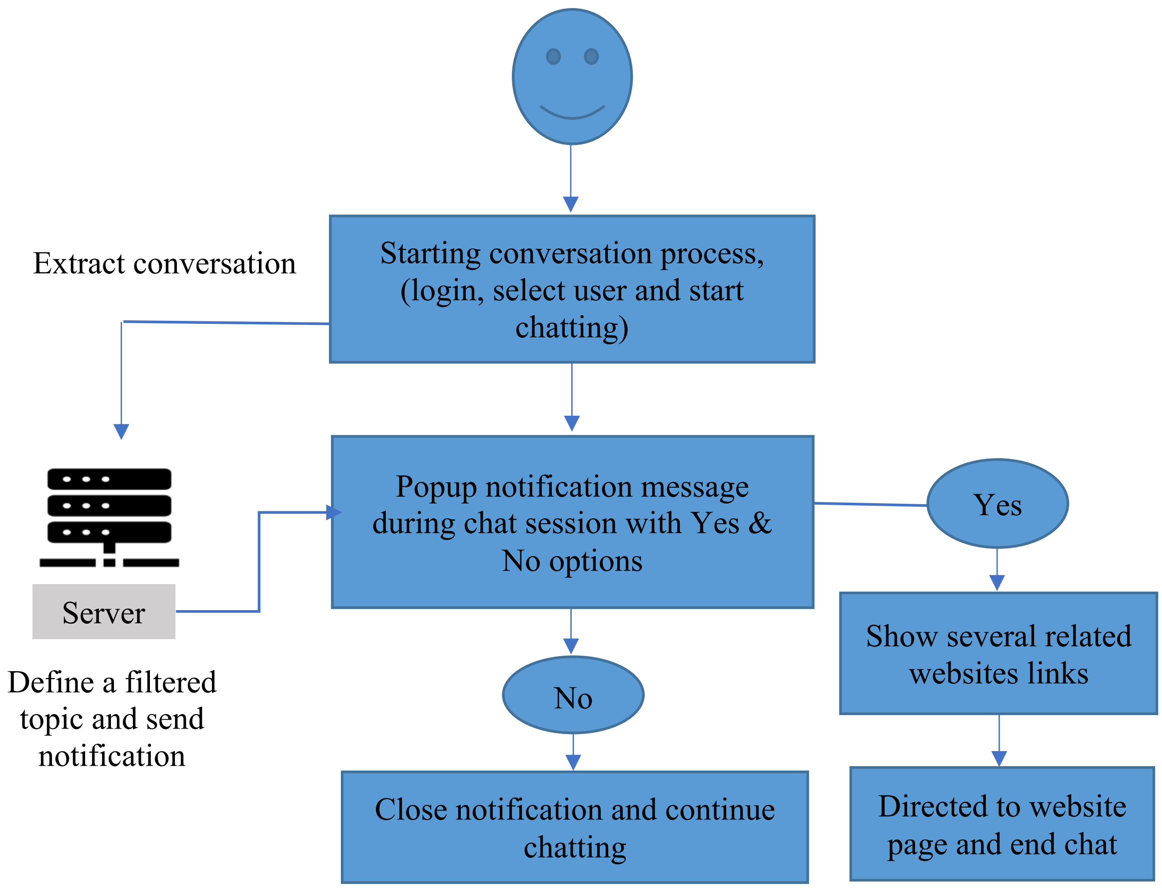


Figure 4.5: The subscriber state diagram.

We used Google's Mobile ad network called AdMob as an advertisement database in our ChatWithRec SRS. We also created our advertisement dataset that contains some popular and widely used websites (3 for each included area) to be suggested to interested users so that they may browse according to their preference. By clicking on the link, a user can leave the conversation window and return later. The travel websites that we included are Expedia[1], Kayak[2], and Trivago[3].

---

https://www.expedia.ca/?pwaLob=wizard-hotel-pwa-v2
https://www.ca.kayak.com/
https://www.trivago.ca/

Further incorporated sites in our proposed ChatWithRec SRS, are Foodora[4], Menu.ca[5], Just Eat[6], Hotel.com[7], Trip Advisor[8], and Booking[9]. In our future work, we will widen our range to incorporate more subjects (e.g., entertainment).

Furthermore, we decided to display the recommended ad as a banner, not an interstitial ad, to ensure that the conversation flow was not interrupted, thus avoiding any disturbance or annoyance to users. Finally, both the user's profile and item descriptions are defined as a key point in any SRS to recommend items that the seeker will probably like. The recommendation system may also produce recommendations with no prompting. Hence, we may increase the number of advertisement categories and sites in order for the advertisement to be more accurate and practical for users.

4.3 Summary

In this chapter, we have presented an overview of our proposed ChatWithRec SRS, including its main details, layers, and technologies that we obtained based on other existing SRSs architectures mentioned before in this work. We also explained the main concepts of each component in our proposed ChatWithRec architecture which are, the online social networks layer, the ChatWithRec app layer that was developed as a proof of concept of our suggested system architecture, and the personal assistant server layer that includes three fundamental modules (conversational analysis module. advertisement recommendation module. and the user profile module). In the next chapter, we will present the implementation details of our proposed ChatWithRec SRS, including its server/mobile intelligent sides, functional/ non-functional requirements, and related technologies that we expended.

https://www.foodora.com/
https://menu.ca/
https://www.just-eat.com/
https://ca.hotels.com/
https://www.tripadvisor.ca/
https://www.booking.com/

Chapter 5 System Design and Implementation

Nowadays, technologies are improving to influence mobile commerce efficiently in several ways such as with its applications, factors, and services without the limitations of time or place (anytime and anywhere). As well, academic and business areas developed in the last few years have provided new information and communication technologies in e-commerce which have impacted the business environment in positive ways. According to (Lee, 2008) the mobile phone will become the essential connection instrument to the internet by 2020 and mobile technologies popularity is increasing gradually and is expected to play a larger role in the business environment, with over half a billion customers following the trend to shop via mobile devices in the coming years. Also, Smartphones nowadays have merged into heavy applications and can augment their capabilities with unlimited computing power and storage space by offloading some services to run on the cloud service side, consequently, saving time and computational power (Kumar. K, 2010; Mell. P, 2011).

Xinwen Zhang (2011) has stated that many available SRS designs enable elastic social network applications to be launched on a mobile device or in the Cloud by using multiple factors, including cloud and device status, application performance measures, and user preferences. One of our goals is to define how to improve mobile chat apps' usability to help users reach the items and topics of their interests. However, recommendation architecture that relies only on mobile applications consumes most of the mobile memory, battery, and other functional and computational resources.

The main contribution of this Chapter is to explain in detail the design and implementation of ChatWithRec SRS (server/mobile intelligent sides), including its functional/non-functional requirements and related technologies that we used based on other existing SRSs architectures mentioned before in this thesis. Hence, our proposed architecture can be simply divided into a group of services executed locally on the mobile side and some other intensive services executed on the Cloud side. Furthermore, many recommendation methods that are used in SRSs don't suffice because they mostly used item/user information without taking into account the user's present context, needs, and could not be extended and applied to other recommendation approaches. To mitigate these issues, our architecture takes into account the relative user's interests and context. The following section includes an exploration of some details about both of the mobile/server design approaches to be adopted in our development architecture; as we mentioned before, we decided to employ both of the (mobile/server) intelligent systems equally.

Some major mobile/server intelligent architecture design approaches are the following:

- Employing everything in the intelligent system (mobile/server) on the OSN's side. The mobile agent relies on the web in this approach and all functions operate within it, for example, extracting the user's conversation, analyzing conversation context, proposing content-related actions to interested users, as well as ad recommendations. This approach is more common and faster to implement, not to mention being a cheaper kind of app development. However, having both intelligent systems (mobile/server) at the same time causes many issues. These issues include making the server massive and expensive to run and not providing a real-time result, since it is based on the cookie method. It is necessary to provide a suitable internet connection in order for it to operate. Hence, it may work under the offline mode but provide only limited functionality using cached pages.
- Employing the intelligent system on the mobile side, providing a more powerful and useful intelligent mobile system and offering access to the system's features and content even without an internet connection. On the other hand, this approach is slow, involves a high cost, and takes a longer time to implement.
- Employing both the intelligent server and the intelligent app separately is certainly useful and more efficient in comparison to the two previous approaches because it provides simple, straightforward, and light app software that can handle many simple functions, as well as a powerful server that can handle the primary system's functions.
- Employing the intelligent system separately, like in the previous approach, divides all the system's functions equally between the server and mobile intelligent sides. The intelligent mobile app is able to work as a complex app and we are able to rely on it if the server fails, goes down or faces any other issue.

In our development architecture, we have chosen to apply both the intelligent server and the intelligent app separately. This provides a light and straightforward software app and an efficient server that runs separately. The server will contain several tasks, which are analyzing the user's conversation context, understanding and defining the conversation session topics, filtering them and matching topics with suitable advertisement recommendations, then sending it to the user's app through the internet to display it. The architecture consists of two components, the mobile-side (client-side), and the cloud-side (server-side). More details about our system architecture are described later in this chapter.

Moreover, various existing SRSs used the CB or CF methods based on an offline data gathering system. Therefore, the real-time data does not consider many current social applications in order to recommend proper items to a user dynamically. In this respect, we aim to design and develop an intelligent real-time SRS which helps to close existing gaps in many related works. The proposed system was developed to recommend advertisements to customers by using their real-time data (conversation) and hybrid filtering methods. The first approach was the CB approach. It follows the rule-based ML method to obtain knowledge about a customer's preferences and to build on the CB profile provided (AJ. Morais, 2012). The second approach uses contextual CF to maintain the customer's satisfaction with the system. Recommendations and suggestions must be quick and delivered during the task session and must update quickly. Figure 5.1 consists of the two main filtering components in our ChatWithRec system.

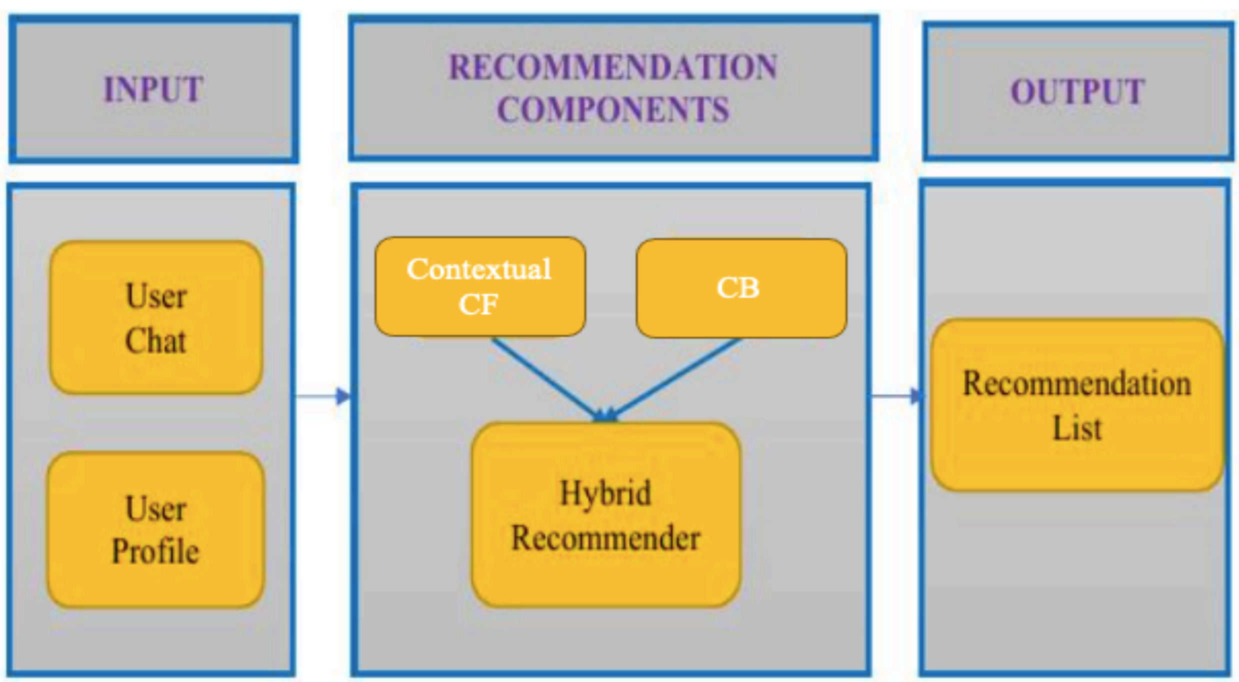


Figure 5.1: General outline of the proposed hybrid real-time SRS.

Current SRSs include two types of techniques of profiling; static and dynamic. In the static technique, all user data and mobile contexts are stored in the system's local database to use later for many tasks such as decision-making. The dynamic technique includes all of the user's data and mobile contexts are captured at runtime then forwarded to the decision engine to make decisions. The dynamic technique provides better decisions, since the mobile contexts change over time such as network bandwidth.

5.1 The Intelligent Mobile-Side

The mobile side is considered the main communication between the users and the server since it handles user requests and display the results. It provides the means to extract conversation sessions and display results (ads recommendation in our work) to users. Understanding the user personality mobile application usage is a fundamental key in current SRSs. The intelligent mobile apps study the user's preferences based on the user's past and real-time interaction data. However, mobile app users are frustrated with mobile app platforms that forcibly push information on to users (promotional content).

The mobile intelligent side should collect realistic data of the mobile device at runtime. The gathered data is used as input to analyze in the conversational analysis module. It also includes other mobile processing tasks, such as task data size per session and mobile battery level. This mobile layer is responsible for retrieving the current location of the user and his/her preferences and sending them to the server-side of the SRS, then displaying the recommended items to interested users.

In addition, multimedia applications require a high level of computing power and are restricted by the limitations of mobile processing power and the battery life of mobile devices. As a result, code offloading is a common solution that is used to overcome mobile hardware limitations. However, it is not convenient to offload the entire application code as some parts of the code need to use other local sensors like GPS. Numerous works have been published on the code offloading method because of the capability of an application to either run locally on a mobile device if there is no connection or to offload some of its computationally expensive code to be executed outside the mobile device when a connection is available like in (S. Kosta, 2012) and (B. Chun, 2011).

Many works discuss different approaches to delivering mobile advertisements. (Gao, 2006) discussed two basic approaches, pull-based and push-based mobile ad delivery. Also, (Tripathi, 2006) discussed a decision policy for delivering advertisements on mobile devices. They looked at the advertisement delivery problem from the perspective of an advertising firm. This firm delivers ads on behalf of its clients, to the mobile customer using a wireless carrier's infrastructure. The ad delivery depended on the customer's information like their real-time location, the available network capacity, and the fee structure agreed upon for ad deliveries. In our design phase, we built an application that could handle many of the computational operations so the mobile device could run as a thin-client. For example, it would display received data from the cloud side then recommend suitable advertisements to the interested user.

5.2 The Intelligent Server-Side

This is the back-end of the system, which provides virtually unlimited hardware and software resources (P. Mell, 2010). It offers infrastructure, a platform, and software as a service to its users. the cloud-side is considered to be the core layer of the proposed system, since it contains the main components that collect cloud server processor speed and utilization rate and pushes them to the mobile device. The server side computes the most relevant recommendations and communicates them back to the mobile device based on the input from the mobile computing layer. The integration and operations of the mobile and cloud computing technologies are enabled through the continuous availability of the network, which enables inter-layer communication. By using the layered architecture pattern, we distinguish between the two distinct concerns of user interaction and system processing with systematic implementation of the RS.

There are numerous tasks that take place on the cloud-side, such as the NLP process (extract, process, and analysis of the metadata) according to the user's requests. It performs several tasks. which are deep analysis of the user's collected data, understanding and defining the conversation session topics, and filtering and matching topics with appropriate advertisement recommendations. Lastly, recommendation will be sent to the user's app through the internet to display. We used the Node.js[1] server platform to develop our system, which is built on Chrome's JavaScript runtime for easily building fast and accessible network applications.

---

https://node.js.org/

From an implementation point of view, the Node.js server interacts with the app through Socket.io, which is a JavaScript networking library that runs server-side on Node.js and in the browser. It also includes convenient features such as broadcasts and multicasts, which are beyond the features of plain web Sockets. By using the Socket.io interface, multiple users can create and send messages as well as receive messages sent by other users. The user needs to be registered and subscribed to the ChatWithRec system to get any recommendations and the user information is stored in the user profile DB. The user's current location and preferences are sent to the Node.js server through Restful API. Figures 5.2 and 5.3 illustrate the system diagram and the mobile-side and server-side agents interaction, respectively.

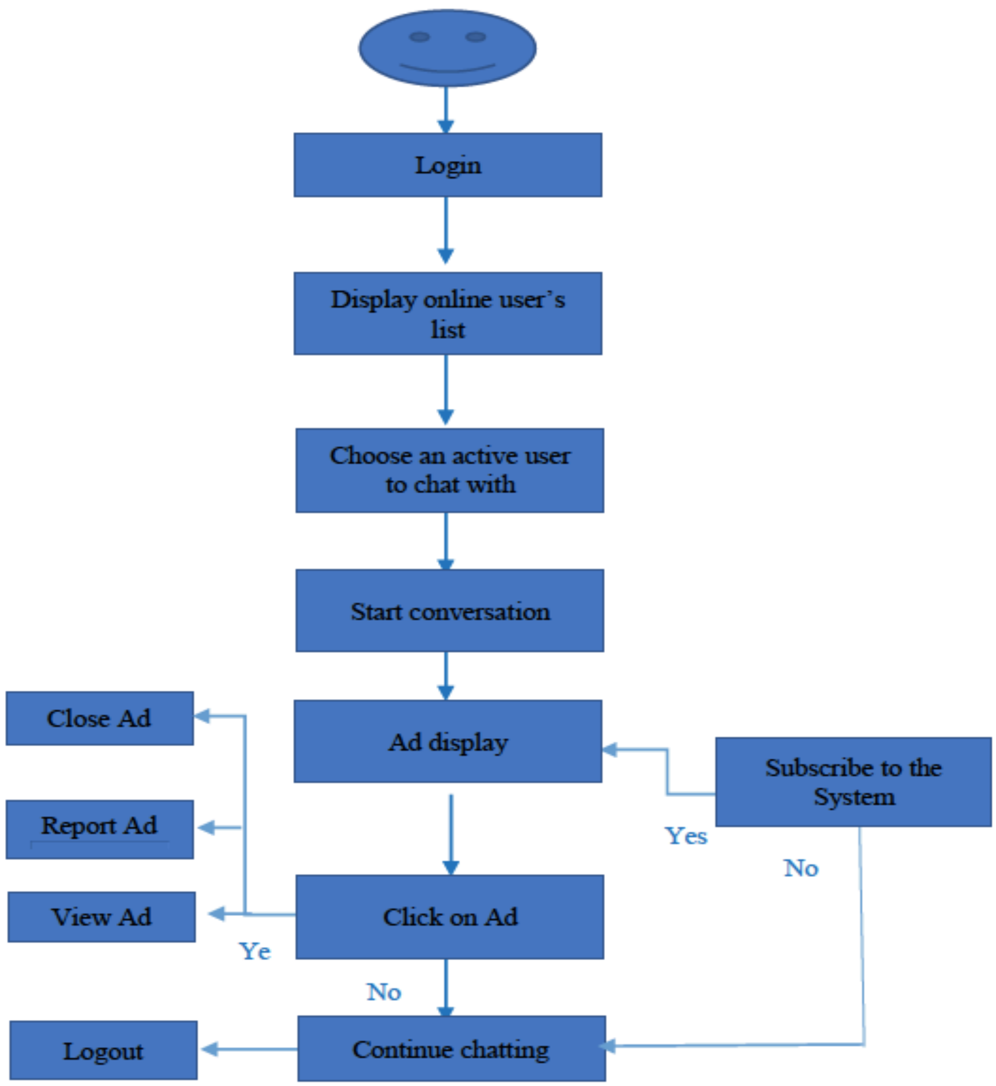


Figure 5.2 : The subscriber state diagram.

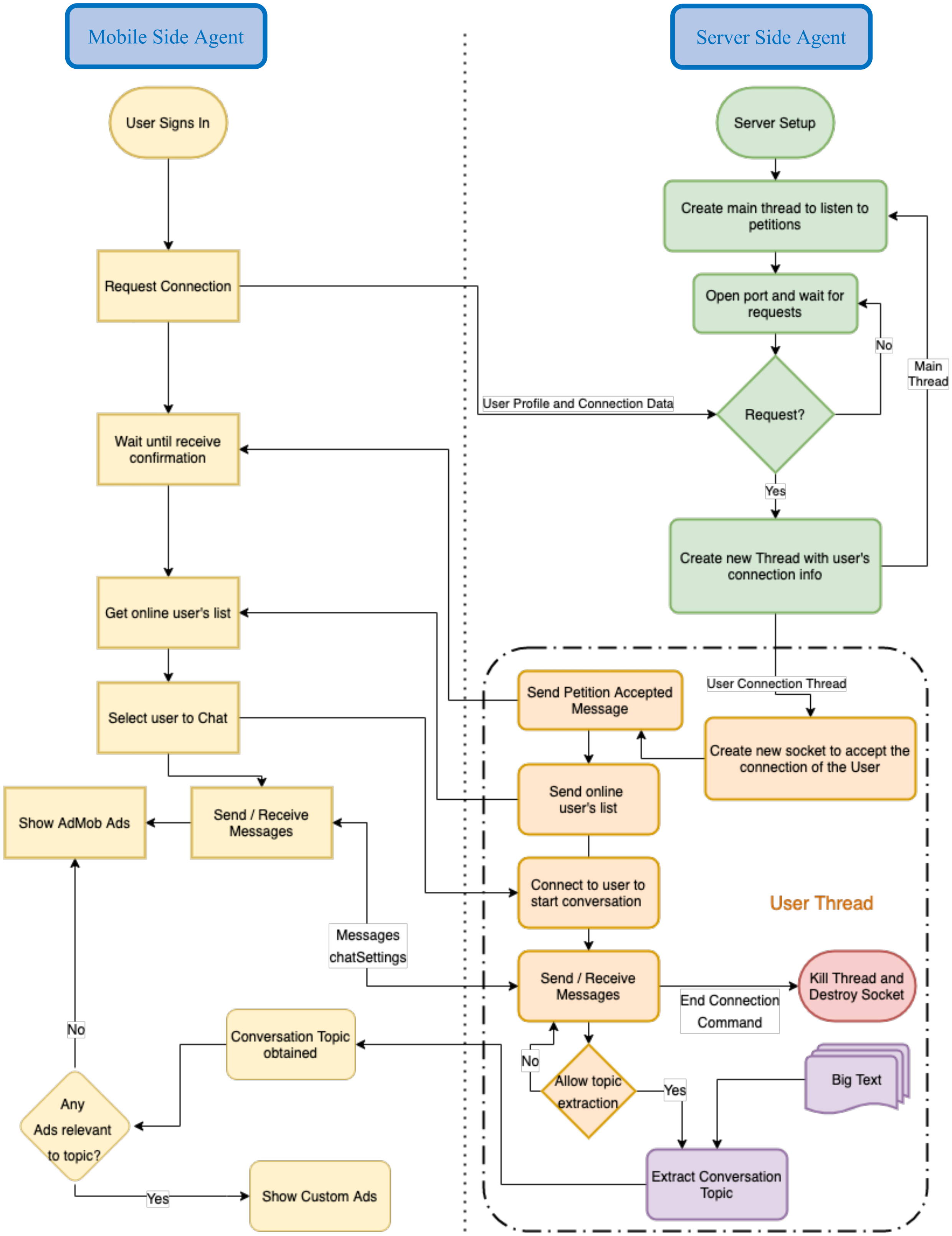


Figure 5.3: The mobile-side and server-side agents interaction.

Technically, the ChatWithRec system's workflow can be divided into the following: analyzing the user's conversation content dynamically, understanding and defining the conversation session topics, filtering them and matching topics with appropriate recommendations such as advertisement recommendations, then sending ads to the user's app to display them to the subscribed and interested user as mentioned in Chapter 4. After recommendations are generated, the system stores and makes them available at the platform's demand for further analysis, like tracking the user's long/short behavior, which we did not consider at this stage of our research.

The Figure 5.3 presents an overview of the system interaction between the client and server intelligent sides. The Node.js server interacts with the app through Socket.io. Essentially, users can easily download the ChatWithRec chat application and register by providing a nickname which will be stored in the user profile as a user ID beside the device ID. We then we get permission from the user to detect and analyze his/her online textual data. Consequently, only subscription users can get the benefit of using our system to receive more personalized and immediate recommendations.

After extracting the user's online conversation we applied the data preprocessing step to clean up the data. The preprocessing steps are supported in Stanford's NLTK Library and contain the following patterns: stop-word elimination, normalization, stemming, lemmatizing, tokenizing, and identifying N-gram. Then we run a modified LDA topic modeling method by applying the following steps: (more details are available in Chapter.6)

- Transform the dialog into a corpus
- Export this corpus to a document-term matrix; the terms are stemmed and the stop words, punctuation, numbers, and terms shorter than three letters are removed.
- Both BoW and TF-IDF algorithms are the first phase in our text mining pipeline, which converts raw text into numerical features that are later fed to the LDA topic model as vectors.
- The BoW assumption counts words that occur many times in a document, while the TF-IDF quantifies the signature list of words from the BoW output (high term frequency = higher importance). We only included terms with a TF-IDF value of at least 0.1, which is a bit less than the median, to make sure that the most frequent terms were omitted.
- Apply the Geolocation API technologies and user's preference topics as a post-filtering step.

- Match topics with ad vectors by pairing data by using a flexible strategy match technique.
- Send a notification message to the subscribed users to determine if we detected the right topic or not. If we were successful, a motivating message will be sent to the user regarding the defined topic, then the recommended advertisement. On the other hand, if the system couldn't detect the right topic, the user can continue his/her chatting and our server will continue to analyze the content to suggest a different topic in real-time.

Also, we included both the AdMob and our customized advertisement database to make sure that we could provide a more accurate recommendation to the user based on his/her current location and context. Finally, the conversation session will be destroyed after the user ends the conversation and logs out of the app to ensure the user's privacy. More details can be found in Chapter 6.

5.3 Functional Requirements

Functional requirements define the fundamental system behaviour (what a system is supposed to do) such as data input, analysis, calculation, software, and hardware specifications, system workflows and authorization levels. Our ChatWithRec system includes two main sections which are:

- Client Application

The client application is the link between the user and the server application. It contains a graphical user interface (**GUI**), gathers information (text and advertisement's preferences) from the users, and allows users to choose ads, handles and manages the user's interactions with the app, and displays advertisement recommendations. Each client is a mobile application that is used to deliver the recommended advertisements and to define the user's location and preferences, Geolocation API technologies are used to define the user's current location.

- Server Application

The server application on the other hand, acts as a back-end system for the recommendation process. It receives information from the client application and is responsible for the data persistence (collected online user-generated content), analyzes the collected conversational session, then provides the client application with related ad recommendations in a real-time manner. The server applies both the CB and contextual CF since it is capable of providing recommendations by interpreting the user content, evaluations and contextual information given by the users. The communication process between the server and the mobile platform is accomplished through an API, developed based on REST principles, due to the advantages and increasing usage of this architectural style.

5.4 Featured Requirements (Non-Functional)

Featured requirements define the system features and overall characteristics that affect the user expectations and experience, such as usability, security, accuracy, and performance. In this thesis, we didn't consider the security aspect since we didn't integrate our system with any of the existing OSN platforms. Instead, we built our own system environment, including its own app and server, which is private and under our visualization.

- Accuracy

The accuracy of recommendations is an important factor, since its improved SRSs and the social context (Chen, Wang, Shi, Feng, & Chen, 2019; Lai, Lee, & Huang, 2019). In our work, the server application creates accurate recommendations that match the user's current request and advertisements that match his preference.

- Usability

The usability factor is considered one of the fundamental factors that expresses the success of a smartphone (R. Baharuddin, 2013). The International Standards Organization (**ISO**) defined the usability factor as the "extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency, and satisfaction in a specified context of use," as shown in Figure 5.4 (S. Ben, 2005).

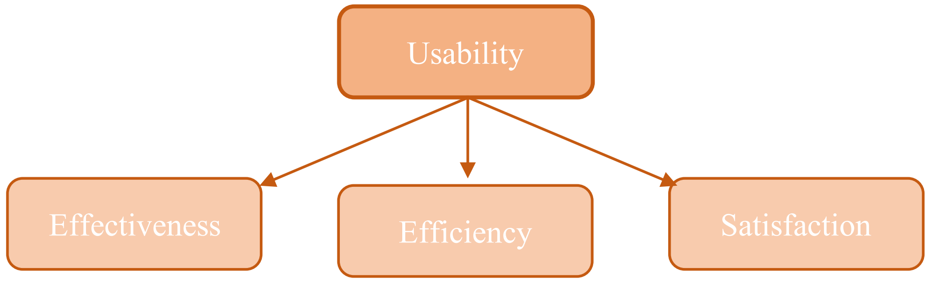


Figure 5.4: The IOS usability model.

5.5 Related Technologies

One of our goals is to define how to improve the usability of mobile chat apps. The mobile chat app is designed to help users reach their interests, items, topics that will enable the user to find the app useful and easier to use. Based on the requirements we mentioned above, numerous common technologies are chosen to support the development of the ChatWithRec SRS which are the following:

- Xcode Environment

Xcode is apple's official Integrated Development Environment (**IDE**) to write software for Mac and iOS developers and software engineers. It allows the users to write software, compile, debug, load into the device, submits to the app store, and other software tasks. Xcode can pull all the tools needed to produce an application into one software package rather than leaving them as a set of individual tools connected by scripts. It also has syntax coloring for programming languages such as Python and HTML. Hence, in our work, we used both Python and Swift programming languages in the Xcode-10.2 version to build our iOS mobile operating system.

- Swift

Swift[2] is an Object-oriented programming language that is used to build software, mobile applications, and servers. We adopted swift language to build our ChatWithRec SRS because it is

https://developer.apple.com/swift/

type-safe, fast, easy to use, easy to add new features. In our App, each page of the ChatWithRec App has its functionality where it gets the data from the registered user's device.

- Node.js

Node.js is an open-source and cross-platform that is built on Chrome's JavaScript running environment and allows JavaScript to run on a server. It is used for fast, quick responding, easy to extend, and accessible network server and web applications. "Node application rely on big processes with a lot of shared states and offers developers a powerful way to develop networking applications that will perform well compared to mainstream solutions" (Rauch, 2012). The Node.js interacts with the app through Socket.io, which is a JavaScript networking library that runs server-side on Node.js and in the browser. It also includes convenient features such as broadcasts and multicasts, which are beyond the features of plain web Sockets. By using the Socket.io interface, multiple users can create and send messages as well as receive messages sent by other users. Node.js can optimize throughput and scaling of mobile applications; indeed, these features are useful in real-time systems so it is the first choice to build our SRS.

- DigitalOcean

DigitalOcean[3] is a popular open-source cloud computing vendor. It has many advantages such as its, independent, low coupling, self-describing, good control panel, and provides a common mechanism for integration between various heterogeneous systems. We hosted our server in the DigitalOcean where it is running continuously to collect online social text from mobile applications (can collect social data from other OSNs).

- Google's Mobile ad network (AdMob):

AdMob is Google's mobile advertising platform that is used for promoting and monetizing mobile applications. It uses the Google mobile ads SDK that aids app developers to understand and gain insights about users as well as maximize ad revenue by collecting information like devices.

---

https://www.digitalocean.com/?refcode=e8a7842ff717

- Gensim Toolkit

An open-source vector space modeling and topic modeling toolkit implemented in Python to lever large unstructured digital texts, it aimed at automatic extraction of semantic topics from documents, using data streaming and well-organized incremental algorithms, which differentiates it from other toolkit software packages.

- Python

It is a high-level object-oriented programming language that is suitable for many application developments besides connecting multiple components. “Python has been preferred because it enables more general-purpose applications while supporting a variety of ML and Deep Learning libraries that facilitate the development of sophisticated data models. Python's readability and performance allow for faster prototype and product development” (Karlijn Willems, 2015). Also, Python includes many ML libraries such as, NumPy and Pandas (Wes McKinney, 2019) that are useful for integrating multiple database content for data analysis.

- Topic Mining Model

TM methods have been established for text mining as it is hard to identify topics manually, which is not efficient or scalable due to the immense size of data. Various TM methods can automatically extract topics from short texts and standard long text data. Such methods provide reliable results in numerous text analysis domains. However, many existing topic modeling methods are incapable of learning from short texts. Also, many issues exist in topic modeling approaches with short textual data within OSN platforms, like slang, data sparsity, unstructured data, non-meaningful and noisy words. In general, TM has proven to be successful in summarizing long documents like news, articles, and books. Conversely, the need to analyze short texts became significantly relevant as the popularity of microblogs, such as Twitter grew. The challenge with inferring topics from short text is that it often suffers from noisy data, so it can be difficult to detect topics in a smaller corpus. Topic mining is a mature area and there are several mainstream algorithms we have used in Chapter 6, such as LDA, NMF, PR, LSA, and PCA unsupervised TM methods, these methods have their features to work in different situations. The fundamental steps involved in text mining are shown in Figure 5.5.

Figure 5.5: The steps involved in a text mining process (Amandeep, 2019).

5.6 The User Cold-start problem consideration

During our implementation, we considered two types of the user's cold-start problem scenarios, one of which involves the new users, the other of which involves existing users. These situation lead to two kinds of recommendation scenarios (cold/warm recommendation) as illustrated in Figure 5.6.

- Pure recommendation model (New User)

Represents the situation when a new user registers, subscribes, and utilizes the ChatWithRec app for recommendations. In this situation, the system only relies on the user's online generated content and location to generate the recommendations without considering his/her preference.

- Hybrid recommendation model (Existing User)

Represents the situation when an existing user utilizes the ChatWithRec app for recommendations. In this situation, the system uses the user's online generated content, location, and past preferences to generate the recommendations.

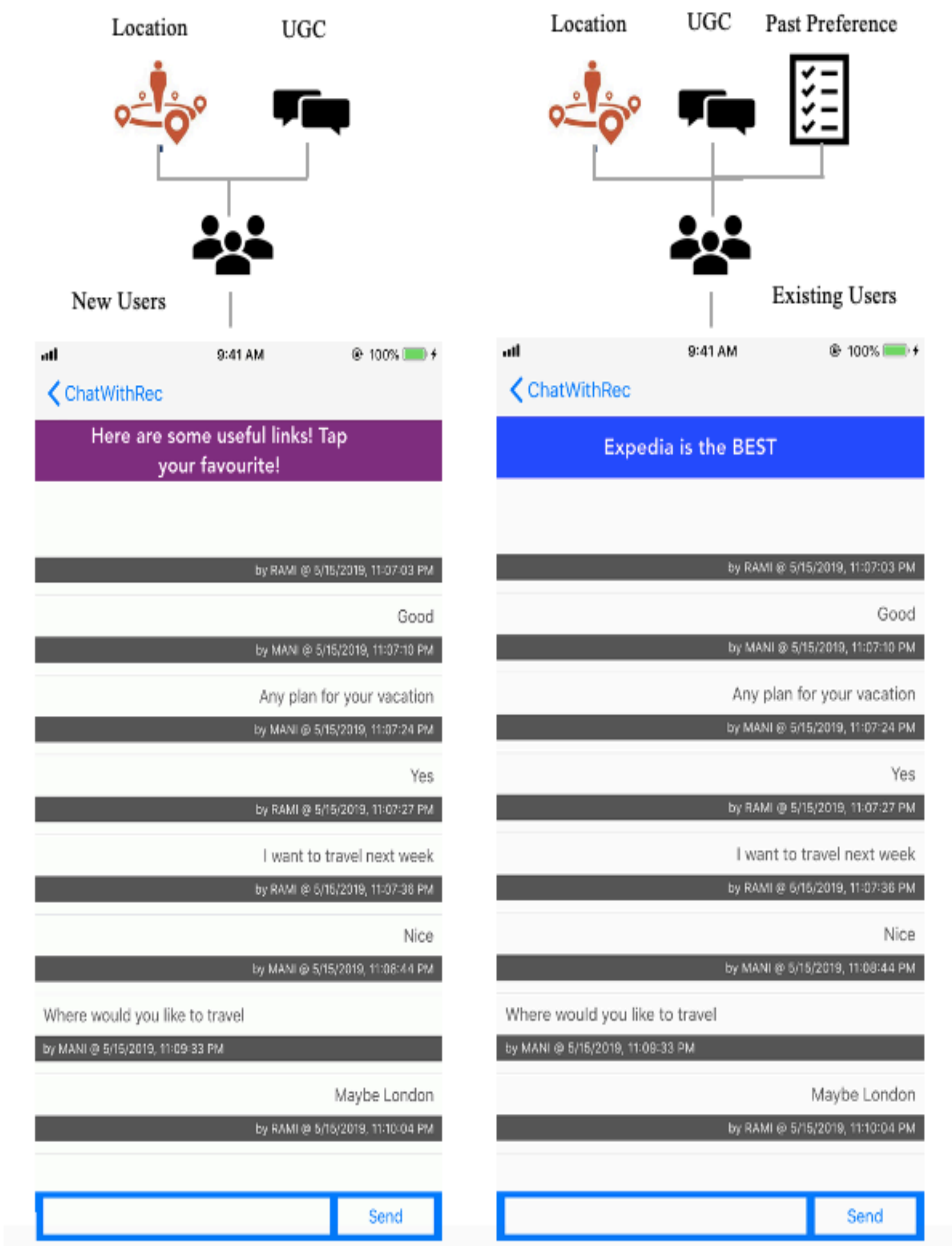


Figure 5.6: Overview of the Source for Implementing Product ommendations.

5.7 Summary

In this chapter, we have explained the design and implementation details of our proposed ChatWithRec SRS, including its server/mobile intelligent sides, functional/non-functional requirements, and related technologies that we used based on other existing SRS architectures mentioned before in this work. Ultimately, we decided to employ both of the (mobile/server) intelligent systems equally. In the next chapter, we will present the evaluation of our developed system and its results.

Chapter 6 ChatWithRec System Evaluation

In previous chapters, we proposed an application architecture for a real-time social recommendation system that aims to improve the performance and accuracy of recommendations, our architecture also presents a possible solution to the user cold-start problem in a real-time manner. In this chapter, we aim to validate the performance and effectiveness of our ChatWithRec social recommendation system as well as to define the user's satisfaction toward their delivered recommendations. The evaluation of our proposed social recommendation system architecture includes two experimental categories (quantitative and qualitative evaluations). The first category focuses on the system's ability to detect the actual topic from the user-generated content extracted from the user's online social networks platform, while the second evaluation category focuses on verifying the effectiveness of our proposed ChatWithRec real-time social recommendation system.

We also created a social network application (chat app) as a proof of concept to verify our developed SRS architecture at this stage of progress. Users can log in to the ChatWithRec app by providing a username and later, they can enter the online user's list and easily start a conversation with another user. Hence, only subscribed users can receive related recommendations that are most interesting to them and that meet their current needs and preference.

Furthermore, we utilize a set of conversation dialogues from several OSNs such as the Facebook platform to test the conversational analysis segment using many topic modeling methods and considering different numbers of extracted topics and features in order to determine which category can give us a clear understanding of online conversation sessions and accurately detect topics. We considered different numbers of topics, t which will provide us with four different groups of topics to compare later in our evaluation based on selected common measures, specifically coherence score, precision, recall, and F-score (Esmaeili et al, 2012).

Following are the contribution of this chapter:

- Investigate select topic modeling methods that are commonly used in text mining, namely LDA, LSA, NMF, principal component analysis (**PCA**), and random projection (**RP**). As there are many TM methods in the field of short textual data, and all certainly cannot be mentioned, we selected the most significant methods for our work.

- Evaluate all included topic modeling methods based on two dimensions: the understandability of extracted topics (topic quality) besides the topic performance and accuracy by applying common standard metrics that apply to the topic modeling domain. These include recall, precision, F-score, and topic coherence. We consider several textual datasets (Authorship blog corpus, Wikimedia articles, 20-Newsgroup and 20-Facebook conversations datasets) that are common for evaluations in SM text application tasks.
- Compare and evaluate many TM methods to define their effectiveness in analyzing short textual social user-generated content.
- Run the completed ChatWithRec SRS with real users to check the proposed system's usability.
- Evaluate user acceptance and satisfaction from using our proposed ChatWithRec system (real user judgments).
- Evaluate two advertisement databases (Google's Mobile ad network called AdMob and our created advertisement dataset that contains some popular and widely used websites to be suggested to interested users so that they may browse according to their preference as a real-time output recommendation mode.
- Evaluate the real-time push notification and ads recommendation to define the correctness and effectiveness of our ChatWithRec real-time SRS (considering its relevantly, and timely).

Many previous studies developed numerous NLP techniques and methods to predict users' keywords, interests, and priorities; however, they were mainly based on importance, word frequency, topical similarity, and rating. It is essential to mention that we are dealing with chat as an online input data source (informal and short dynamic sentences). Topic modeling is a methodology for processing the massive volumes of data generated in OSNs and extracting the veiled concepts, protruding features, and latent variables from data that depend on the context of the application (Pooja. K. 2018). Several methods can operate in the areas of information retrieval and text mining to produce keywords and topic extraction such as MAUI, Gensim and KEA, etc.

Below are the toolkits that commonly used in NLP for topic modeling:

- **TextRank,** proposed in (Mihalcea, 2004), that is a common an unsupervised graph-based approach used to extract significant keyword and keyphrases from documents, indeed, TextRank inspired by famous PageRank algorithm (Brin and Page, 1998) and motivated many further researches like (Liu, 2010). Researcher (Hulth, 2003) applied a supervised learning system to extract a keyword from abstracts since she believed that it's more beneficial than using the full text. The reader cannot access the entire document because just the abstract is available to read in numerous online documents.
- **Stanford Topic Modeling Toolbox (TMT),** presented by (Ramage, 2009), was written by the Stanford NLP group. It is designed to help social or other researchers who aim to analyze large textual material. It includes many topic algorithms such as LDA and the input can be text in Excel or further spreadsheets.
- **KEA,** is open-source software distributed in the Public License GNU and used for keyphrase extraction from the entire text of a document; it can be applied for free indexing or controlled vocabulary indexing in the supervised approach. KEA was developed by using the Java language; it is a simple and efficient two-step algorithm that can be used across numerous platforms.
- **Machine Learning for Language Toolkit (MALLET),** is first released in 2002 by (Mccallum, 2002) MALLET[1] is a TM tool that is written in Java code for application of machine learning like NLP, document classification, clustering, and information extraction to analyze large volumes of unlabeled text. MALLET includes different algorithms to extract topics from a corpus such as Pachinko Allocation and Hierarchical LDA.
- **Gensim**, was presented by (Rehurek, 2010), as open-source vector space modeling and topic modeling toolkit that implemented in Python to lever large unstructured digital texts, it aims to extract semantic topics from documents, using data streaming and efficient incremental algorithms, which differentiates it from most other scientific software packages that only target batch and in-memory processing.

---

https://diliprajbaral.com/2017/06/04/topic-modelling-lda-mallet

6.1 The Dataset

Online UGC contains many irrelevant noise data that are meaningless, irrelevant, and unneeded for NLP algorithms such as meaningless words and symbols which need to be cleaned before applying any text analysis techniques in SRSs domain. We included different databases in our evaluation process, which are short and straightforward data as shown in Table 6.1.

Table 6.1: shows the statistics of our involved datasets.

Dataset	Description
Wikimedia articles dataset[2]	enwiki-20200301-pages-articles-multistream.txt.bz2 16.7 GB
Blog Authorship Corpus[3]	posts of 19,320 bloggers gathered from blogger.com in August 2004. or approximately 35 posts and 7250 words per person.
20 Newsgroup[4] data	20,000 documents Average document length: 28 Topics: computer, recreation, science, miscellaneous, politics, and religion as distinct classes
20-Facebook Conversations[5]	20 text conversations Approximately 87 sentences and 7250 words. Topics: travel, food, restaurant, hotel booking, flight booking, study and university

https://dumps.wikimedia.org/enwiki/
http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm
http://people.csail.mit.edu/jrennie/20Newsgroups/
https://www.facebook.com/roro.albalawi

6.2 DSR Evaluation Metrics

Various evaluation methods are used to evaluate the performance of SRSs, but online and offline evaluation methods are the most applied methods in the evaluation protocols. We follow (Gunawardana and Shani, 2015) in their description of these methods. The online evaluation approach provides very trustworthy results, since it measures performance within the real application on real users; however, it takes too much time and effort when being applied to a real-life dataset, especially since it involves firm requirements on response time, budget, and large user/item bases. On the other hand, the offline evaluation approach is usually used at the beginning of the development evaluation because it is not concerned with a running system. However, the offline evaluation relies on the incompleteness of the dataset.

The goal of evaluating the ChatWithRec SRS is to determine whether or not it detects the topic that the user is talking about in real-time (same conversational session) and whether it recommends interesting items that are related to the user's current need which he might view, buy or listen to. Consequently, in our evaluation process, we measured the predictive power of our proposed ChatWithRec system by using both online and offline evaluation approaches. Moreover, we used many classification accuracy metrics to evaluate the performance and accuracy of our work which are, Precision, Recall, F-score.

In general, the relationship between mentioned metrics is often referred to as a confusion matrix that describes each item and categorizes them into True-Positive, False-Positive, True-Negative, or False-Negative selections. Furthermore, the precision measure is a ratio of the number of efficient recommendations to the total number of recommendations provided to an active user. Recall measure is a ratio of the number of efficient recommendations to the number of user desirable items. The F-measure is the weighted average of the precision and recall, which is calculated by the ratio of multiplication of two numbers to the sum of them. The value of all three measures is in [0, 1] and 1 is the best value.

Hevner (2004) Considered the evaluation process as the main stage of the DSR methodology, which contains many critical terms like usability, functionality, accuracy, performance, and others that can evaluate an artifact. Nevertheless, there is limited direction regarding how and which of these terms and criteria should be applied. DSR contains several evaluation criteria and different ways to be estimated. (Peffers, 2012) presented a survey reviewing

142 DSR articles. They concluded that the most used evaluation methods in the DSR are descriptive scenarios and technical experiments. Chapters 6 and 7 include more details and explanations.

Our evaluated criteria are presented below:

6.2.1 Usability Measured

One of our goals is to define how to improve mobile chat apps usability to help users reach their interests and topics and to be usable and useful and easier to use. In general, the usability factor includes many features such as effectiveness, efficiency, and satisfaction. Effectiveness (demonstrating the level of accuracy and completeness of goal achievement); efficiency (how well resources were used for the sake of effectiveness); and satisfaction (relief, and positive user interaction whilst operating the software). Each one of these factors affects overall how the software will be designed. Specifically, it affects user interaction with the system.

In our work, the user's satisfaction is measured through survey questionnaires that aimed to measure the user's self-reported satisfaction. We created a survey questionnaire that targeted groups of people who were going to participate in our experiment process to measure all presented criteria in our system. The survey questionnaires contained three dimensions to measure user fulfillment: perceived personalization and perceived enjoyment and perceived ease-of-use.

6.2.2 Accuracy Measured

Accuracy denotes how correct and precise the system is and we used many statistical metrics in our experiment to evaluate the accuracy of our ChatWithRec SRS. We chose to use the statistical metrics below, which are considered the most adaptive and standard evaluation measurements to calculate RSs effectiveness.

Recall: (as in Equation.1) is a common information retrieval metric that measures the fraction of relevant items among the recommended items.

Precision: (as in Equation.2) is a common information retrieval metric that measures the fraction of retrieved recommended items to the actual relevant items.

F-score: (as in Equation.3) measure the effectiveness of the retrieval.

$$\text{Recall} = \frac{tp}{tp+fn} \qquad \text{Eq.1}$$

$$\text{Precision} = \frac{tp}{tp+fp} \qquad \text{Eq.2}$$

$$\text{F-score} = \frac{Precision\ .\ Recall}{Precision+Recall} \qquad \text{Eq.3}$$

Note: True Positive (TP) is the number of keywords detected as a topic, False Positive (FP) is the number of non-keywords detected as a topic, True Negative (TN) is the number of non-keywords detected as non-topics, and False Negative (FN) is the number of keywords detected as non-topics.

Coherence Score: (as in Equation.4 and 5) described how interpretable a topic is based on the degree of significance between the words within the topic itself. The topic coherence measures aim to evaluate the quality of topics from a human-like perspective. A discussion of diverse coherence score measures can be found in (M. R¨oder, 2015).

$$C\ (t\ ,\ W_{t)}\ ) = \sum_{w1,w2\,\epsilon Wt} log\ \frac{d\ (w1,w2)+\ \epsilon}{d\ (w1)} \qquad \text{Eq.4}$$

$$\text{Coherence Score} = \frac{tp+tn}{tp+fp+tn+fn} \qquad \text{Eq.5}$$

Where t is topic, W_t is a set of words that described the topic t, d (wi) is the number of documents that have *wi, d (wi, wj)* is the number of documents where *wi* and *wj* co-occur, and ϵ indicates a smoothing factor that usually returns real numbers set to 1 or 0.01.

6.3 The ChatWithRec SRS Initial Evaluation

6.3.1 The TM Methods First Evaluation

In this section, we consider short and straightforward textual data of popular OSNs such as Facebook. Facebook is one of the most popular social networking sites and caters to millions of people who interact with others and share their thoughts, stories, and ideas (Kamal and Arefin, 2016). In Table 6.2, we present the result of one short conversation sample that we extracted from the Facebook platform.

Table 6.2: Shows a single result from our initial topic modeling experiment in the Facebook conversation.

Tools	Extracted Keywords
Stanford	New Jersey, New York, General, Next week, Next month, Travel
Google iCloud	Family, Flower, New Jersey, New York, Park, Fun, Place, People, Thanks, Travel
Cortical.io	Fun, Lot, New York, Love, Travel, Square, Month, Park, Wee, Thanks
Extractor	York, Travel, Time square, Jersey, Planning, Flower, bought
Kea	New York, Good, Travel, Fun, Next, Family, Time square
TextRank	Good time, New York, Next, Week
Gensim	New York, Good, Travel, Time, Next, Family, See, Jersey

6.3.2 The TM Methods Second Evaluation

We then focused on three topic modeling tools, Kea, Gensim, and TextRank, to test both included datasets by comparing the results of extracted keywords and some evaluation metrics which are: Precision, Recall, and F-score. The Gensim tool gets a higher F-score in both datasets. Figure 6.1

shows the results of our second experiment. And Figure 6.2 shows a snapshot of our evaluation environment.

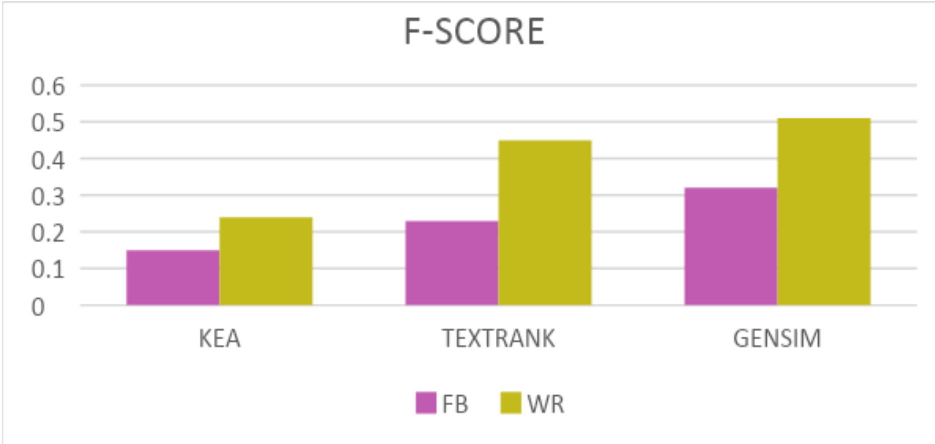


Figure 6.1: The F-score results in all three included topic modeling tools.

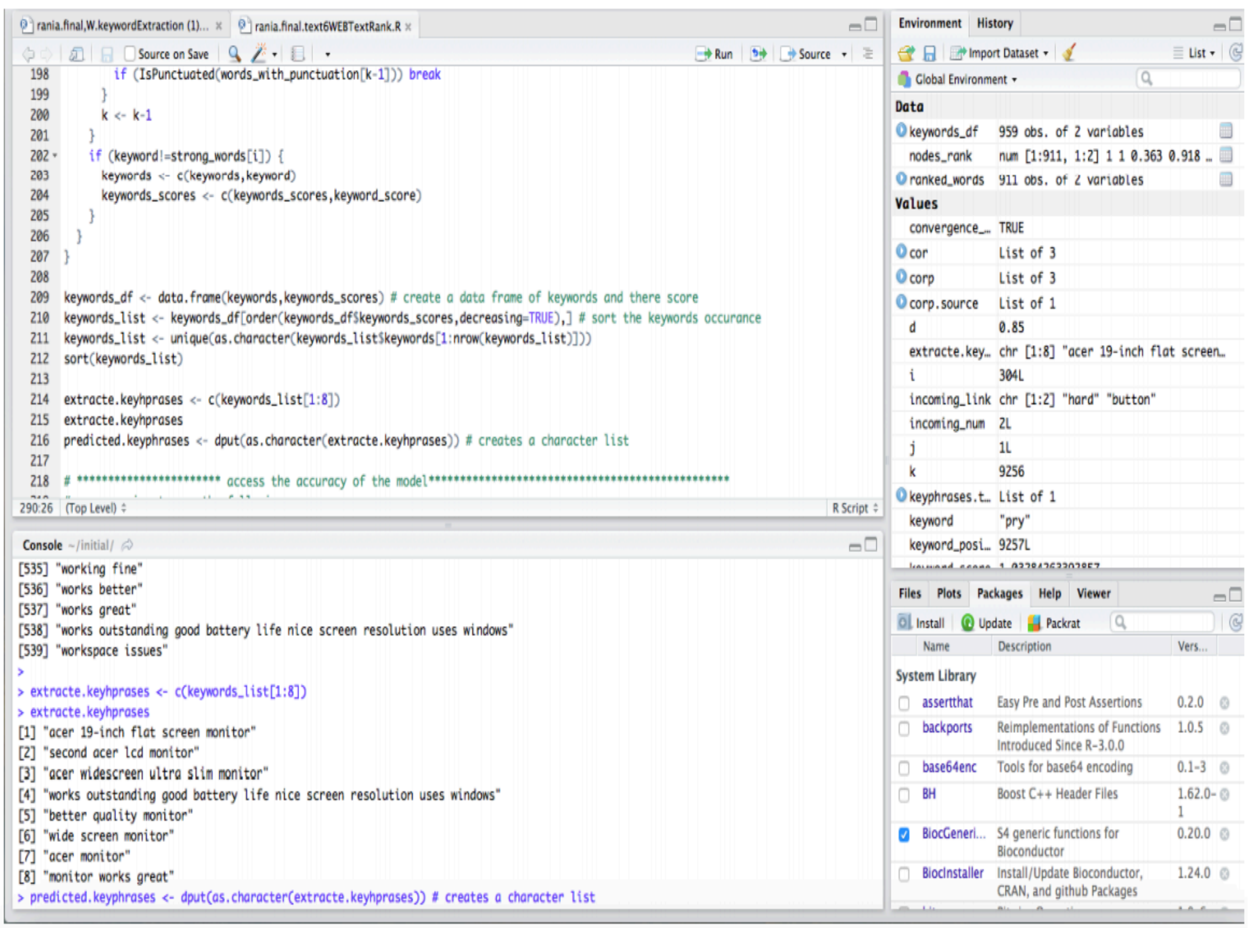


Figure 6.2: A snapshot after running web review data in TextRank tool.

We decided to implement the Gensim toolkit since it gives more accurate results in comparison to KEA and TextRank tools. Gensim was the most popular tool used in many recent studies as it offers more functionality results. We used the Gensim library to implement the BOW, tf-idf, and LDA models since it contains an NLP package with implementations of much well-known functionality for the tasks of topic modeling. Basically, TM is a methodology for processing massive volumes of data generated in OSNs and extracting the veiled concepts, protruding features, and latent variables from data that depend on the context of the application (Pooja. K. 2018). In general, TM methods are used to measure the word frequencies and distance between words that occur in a corpus. Several methods can operate in the areas of information retrieval and text mining to perform keywords and topic extraction such as MAUI, Gensim, and KEA, etc. In our work, we focused on numerous frequently used TM methods that are built using a diverse representation form and statistical models. A standard process for topic generation is shown in Figure 6.3.

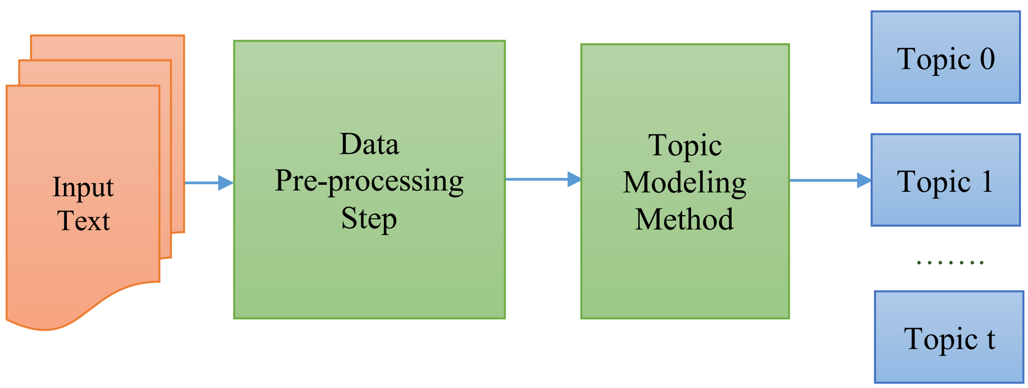


Figure 6.3: Topic modeling for text data.

We evaluate the topic quality and performance of many TM methods. The fundamental difference among all involved methods is in how they capture the structures and in which parts of the structures they exploit change this sentence as it is unclear. There are numerous TM methods used in the field of social media and textual data. Since we cannot mention all of the methods, we compared the most popular ones in order to determine which method is suitable to integrate into our future proposed real-time SRS called ChatWithRec SRS (Albalawi. R, 2019) and (Albalawi. R. Yeap. T, 2019).

In addition, supervised and unsupervised learning are two topic modeling approaches that are used in the NLP tasks. The supervised learning method uses data with labels in order to solve a problem and can utilize side information such as ratings. On the other hand, the unsupervised learning method doesn't need to train or label the data; it also doesn't require any prior knowledge about it. We adopted the unsupervised method since we are dealing with unstructured online data. Hence, the LDA topic modeling method can train the data online, which is useful in our proposed system.

6.3.3 The Topic Modeling Texted Methods

LSA: It is a method in NLP proposed by (Deerwester et al., 1990), particularly distributional semantics, that can be used in several areas, such as topic detection; it has become a baseline for the performance of many advanced methods. Distributional hypotheses make up the theoretical foundation of the LSA method which states terms with a similar meaning closer in terms of their contextual usage, assuming that words that are near in their meaning show in the related parts of texts (S. Dudoit, 2002). Also, it analyses large amounts of raw text into words and separates them into meaningful sentences or paragraphs. LSA considers both the similar terms of text and related terms to generate more insights into the topic. Besides, the LSA model can generate a vector-based representation for texts which aids the grouping of related words. A mathematical approach called SVD is used in the LSA model to outline a base for a shared semantic vector space that captures the maximum variance across the corpus. (Pinaki Prasad, 2020) stated that LSA method which learns latent topics by performing matrix decomposition on the term-document matrix, let's say X is a term-by-document matrix that decomposed into three other matrices S, W, and P, multiplying together those matrices, give back the matrix X with, {X} = {S} {W} {P}, each paragraph is characterized by the columns and the rows characterize the unique words. Figure 6.4 presents the SVD of the LSA topic modeling method.

Figure 6.4: SVD of the LSA topic modeling method. (Pinaki Prasad, 2020).

- **LDA**: is a probabilistic model that is considered to be the most popular topic modeling algorithm in real-life applications to extract topics from document collections since it provides accurate results and can be trained online. LDA model aims to define the mixture of topics that a document includes. Corpus is organized as a random mixture of latent topics in the LDA model, and the topic refers to a word distribution'. Also, LDA is a generative unsupervised statistical algorithm for extracting thematic information (topics) of a collection of documents within the Bayesian statistical paradigm. The LDA model assumes that each document is made up of various topics, where each topic is a probability distribution over words. A significant advantage of using the LDA model is that topics can be inferred from a given collection without input from any prior knowledge. A schematic diagram of the LDA topic model is shown in Figure 6.5.

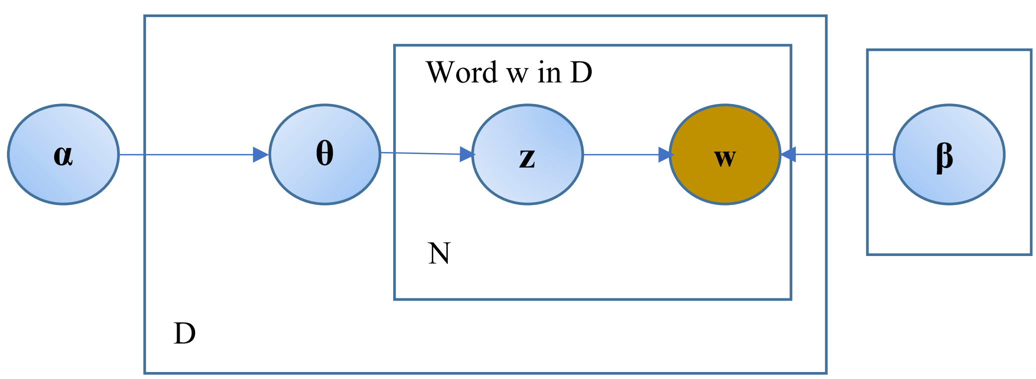


Figure 6.5: The original structure of the LDA topic model.

In Figure 6.5, α is a parameter that represents the Dirichlet prior for the document topic distribution, β is a parameter that represents the Dirichlet for the word distribution, θ is a vector for topic distribution over a document d, z is a topic for a chosen word in a document, and w refers to specific words in N, plate D is the length of documents, and plate N is the number of words in the document.

NMF: It is an unsupervised matrix factorization (linear algebraic) method that is able to perform both the dimension reduction and clustering simultaneously (Jingu, 2014). It can be applied to numerous topic modeling tasks; however, only a few works were reported to determine topics for short texts. (Yan et al, 2013) presented an NMF model that aims to obtain topics for short text data by using the factorizing asymmetric term correlation matrix, the term-document matrix, and the bag-of-words matrix representation of a text corpus. (Chen, 2019) defined the NMF method as a decomposing a nonnegative matrix D into non-negative factors U & V, V≥0, U≥0 as shown in Figure 6.6. The NMF model can extract relevant information about topics without any previous insight into the original data. NMF provides good results in several tasks such as image processing, text analysis, and transcription processes. In addition, it can handle the decomposition of non-understandable data like videos.

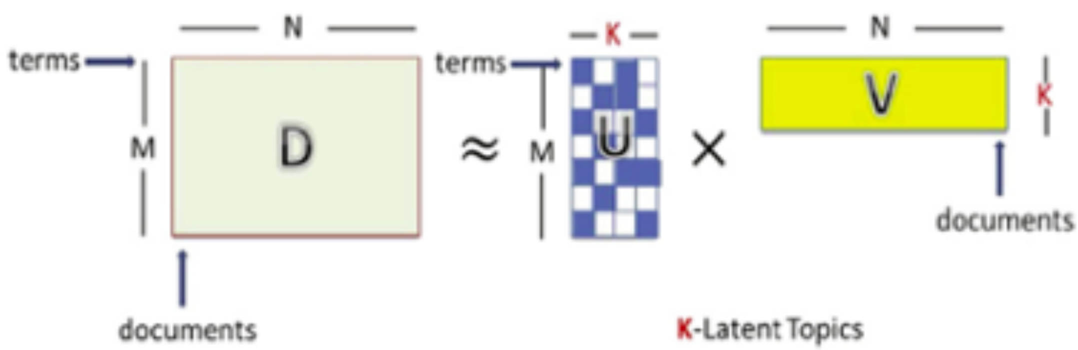


Figure 6.6: The original structure of the NMF topic model.

In Figure 6.6, D≈UV, U and V are elementwise non-negative and for a given text corpus is decomposed into two matrices which are term-topic matrix U, and topic-document matrix V, corresponding to K coordinate axes and N points in a new semantic space, respectively (each point represents one document).

- **PCA**: It is an essential tool for text processing tasks, and it has been used since the early 90s (Jolliffe, 1986; Gomez, 2012). The PCA method has been used to decrease feature vectors to a lower dimension while retaining the most informative features in several experimental and theoretical studies. However, it is expensive to compute for high-dimensional text datasets. “The PCA topic modeling method found a d-dimensional subspace of Rn that could capture as much the dataset’s variation as possible, specifically, given data S ={ x1, …, xm} would find the linear projection to Rd as in Equation 6;

  - $\sum_{i=1}^{m} \| \chi_i^* - \mu * \|^2$ Eq.6

  where χ_i^* is the projection of a point χ_i and $\mu *$ is the mean of the projected data.

- **RP**: It has attracted attention and has been employed in many machine learning scenarios recently as classification, clustering, and regression (Wang. J, 2006), and (Ramage, 2011). The RP topic modeling method uses a random matrix to map the original high-dimensional data onto a lower-dimensional subspace with a reduced time cost. The main idea behind the RP method stems from (Johnson WB, 1984), who states that as “a set of n points in a high-dimensional vector space can be embedded into $k= \vartheta(\epsilon-2 \log n)$ dimensions, with the distances between these points preserved up to a factor of $1+\epsilon 1+\epsilon$. This limit can be realized with a linear projection $\tilde{\mathbf{A}}$ =**AR**, for a carefully designed random matrix **R**$\in \mathbb{R}d\times k$ ($k\ll$d), where **A**$\in \mathbb{R}n\times d$ denote a data matrix consisting of n data points in $\mathbb{R}d$” (Wójcik, P.I, 2019). In addition, RP has attracted lots of attention, and its accuracy for dimensionality reduction of high-dimensional datasets and directions of projection is independent of the data (doesn’t depend on training data). Still, RP delivers sparse results because it does not consider the fundamental structure of the original data and frequently leads to high distortion.

We also define the main advantages and disadvantages of all involved topic methods as shown in Table 6.3, and we evaluate the topic quality and performance of many TM methods;

Table 6.3: Includes the main advantages and disadvantages of the topic modeling methods:

METHOD	ADVANTAGE	DISADVANTAGE
LSA	Solves the data sparsity problem and captures synonyms of words. Reduces the dimensionality of TF-IDF by using Singular Value Decomposition. It doesn't require a robust statistical background and probability theory. Exploits unique structures as factors.	Difficult to label a topic in some cases and to establish a number of topics. The determination of topic numbers depends upon the human judgment and cannot be determined statistically. It doesn't capture the correlation between multiple topics.
LDA	It does not require any previous training data. Provide more semantically interpretable and performs well if there is no time constraint. Handles long-length documents and is able to show adjectives and nouns in topics. Handles mixed lengthy documents. Able to enhance transitive relations between topics and obtain high-order co-occurrence in small documents.	Needs aggregation of short messages to avoid data sparsity in short documents. Unable to model relations among topics that help to understand deeply the structures of documents. A slow process algorithm. Requires a pre-defined number of topics (T), If T is too small= topics are more general if T is too large= topics will be overlapping with each other.
NMF	Fast process for a large amount of real-time data. Able to extract meaningful topics without prior information or knowledge of the underlying meaning in the original data. Appropriate for word and vocabulary recognition tasks.	Sometimes provides semantically incorrect results.

PCA	Low noise sensitivity and decrease the need for capacity. Maintains the best possible estimate and works well on moderately low dimensional data. It decreases the noise data because the maximum variation source is chosen, and the small variations are ignored automatically. Recommended in work that aims to introduce new features by losing original features in the procedure of transformation of the high dimensions data into low dimensions. Delivers an output that can be visualized as a solid version of the main dataset.	The covariance matrix is difficult to evaluate in an accurate manner (P. J. Phillips, 2005). Cannot detect the simplest invariance data sometimes, unless the training data explicitly offer this information (C. Li, 2008). Expensive to compute particularly for high-dimensional datasets.
RP	Robust. Provides good results in data streaming tasks and if data is so high dimensional. Valid to use in imbalanced datasets. Advance linear separability. Good at discovering discriminative features.	Data sparsity. Slow predictions. Sensitive to noise data. Bad at fitting complex features. Applicable to only a few datasets.

6.3.4 Data Preprocessing

In our experiment, all input data was text data that possessed English language properties. The first steps in the text mining process were to collect unstructured and semi-structured data from multiple data sources like microblogging and news web pages. Next, the preprocessing step was applied to clean up the data and then convert the extracted information into a structured format to analyze the patterns (visible and hidden) within the data. Extracted valuable information can be stored in a database, for example, to assist the decision-making process of an organization.

In addition, we utilized the spaCy besides the pyLDAvis libraries for the named entity recognition, stemming, and other data preprocessing processes (M. Honnibal, 2017; C. Sievert, 2014). Corpus preparation and cleaning were done using a series of packages running on top of Python as NLTK –Natural Language Toolkit that provides stop word removal, stemming, lemmatizing, tokenization, identifying n-gram procedure, and other data cleanings like lowercase transformation and punctuation removal (Kolini, 2017; Phand, 2018).

The preprocessing steps are supported in Stanford's NLTK Library and contain the following patterns:

- Stop-Word Elimination: remove most common words in a language that are not helpful and in general unusable in text mining like prepositions, numbers, and words that have not contained applicable information for the study. In fact, in NLP there is no particular general list of stop-words used by all developers who choose their list based on their goal to improve the recommendation system performance.
- Normalization: text transformed into a single basic format by converting the characters to lowercase, deleting all symbols, numbers, punctuation, and not useful words or letters. This step is essential in order to shrink the size of the vocabulary.
- Stemming: the conversion of words into their root, using stemming algorithms such as Snowball Stemmer.
- Lemmatizing: used to enhance the system's accuracy by returning the base or dictionary form of a word.
- Tokenizing: divide a text input into smaller parts (sentences), and the sentences into smaller pieces called (tokens) like phrases, words, or other meaningful elements (tokens). The outcome of the tokenization is a sequence of tokens.
- Identifying N-gram Procedure such as bigrams (phrases containing 2 words) and trigrams (phrases containing 3 words) words and consider them as one word.

After the preprocessing step, the conversion sentences convert into a sequence of words/terms for each sentence that can be used for further processing. Then we applied a commonly used term-weighting method called TF-IDF, which is a pre-filtering stage with all the included TM methods. TF-IDF is a numerical statistical measure used to score the importance of a word (term), in any content, from a collection of documents based on the occurrences of each word. It also checks how relevant the keyword is in the corpus.

Moreover, it considers the frequency of the word and induces discriminative information for all terms. Term Frequency represents how many times a word appears in a document, divided by the total number of words in that document. Inverse Document Frequency is calculated by how many documents the term appears in and divides it by the number of documents in the corpus.

Moreover, "calculating the TF-IDF weight of a term in a particular document, requires calculating term frequency (TF(t,d)) which is the number that the word t occurred in document d, as well as document frequency (DF(t)), is the number of documents in which the term t occurs at least once and inverse document frequency (IDF) that can be calculated from DF using the following formula. The IDF of a word is considered high if it occurred in a few documents and low if it occurred in many documents" (Ahmed. I, 2018).

$$\text{TF} = \frac{num\ of\ occurrences\ of\ word\ in\ documents}{num\ of\ words\ in\ all\ documents} \qquad \text{Eq.7}$$

$$\text{IDF} = \log \frac{num\ of\ documents}{num\ of\ documents\ with\ word\ occurs} \qquad \text{Eq.8}$$

An example:

Suppose we have a document that has 600 words. Suppose the word "Moon" appears 8 times in the document. Then **tf** for "moon" is (8/600) = 0.013

If the corpus consists of 10,000,000 docs and the term "Moon" appears in 2500 of them then **idf** is log(10,000,000/2500) = 3.60

Therefore, TF-IFD = tf * idf = 0.013 * 3.60 = 0.0468

6.3.5 Evaluation Procedure

OSNs include a huge amount of UGC with many irrelevant and noisy data. Examples of this include non-meaningful, inappropriate data and symbols that need to be filtered before applying any text analysis techniques. In our work, we deal with text mining subjects. This is quite difficult to attain because the objective is to analyze unstructured and semi-structured text data. Without a doubt, employing methods that are similar to human-human interaction is more convenient, where users can specify their preferences over an extended dialogue.

Moreover, there is a need for further effective methods and tools that can aid in detecting and analyzing online social media content, particularly for those using online UGC as a source of data in their systems. We implemented the Gensim Toolkit due to its ease of use and because of its more accurate results. Gensim was the most popular tool used in many recent studies and it offers more functionality; it also contains an NLP package that has effective implementations of several well-known functionalities for topic modeling methods such as NMF, LDA, and LSA.

In our experiment, we tested numerous topic modeling methods on commonly used public text datasets for experiments in the text application task called, “The Newsgroup and 20-Facebook short conversation data” from the Facebook social network site. We evaluate the topic quality and performance of five frequently used TM methods. In addition, we calculate the statistical measures Precision, Recall, and F-score to assess the accuracy verification within a different number of features. It is important to consider how many topics we want to extract and find in the corpus. This step must be decided by a human user. All dataset description and equation details were defined before in Chapter 3.

6.3.6 Data Extraction and Experiment Results

During our data extraction stage, we aimed to extract topics from clusters of input data. As we mentioned before we did our second evaluation several times by applying a different number of features f and topics t, f = 10, 100, 1000, and 10000, t = 5,10, 20, and 50. Tables 6.4, 6.5, and 6.6 present our initial results of the topic performance and accuracy after applying some common standard metrics that apply to the topic modeling methods, related to the 20-Newsgroup data.

Table 6.4: Shows the performance of involved topic modeling methods with different extracted topics t, t=5 and 10, (average value of recall, precision, and F-score).

TM Method	**Number of Topics**					
	5			**10**		
	R	P	F	R	P	F
LSA	0.1546419	0.1501913	0.1523841	0.1825729	0.1838501	0.1881104
LDA	0.150000	0.1533333	0.1511765	0.1238715	0.1067887	0.1146975
NMF	0.2577005	0.2522465	0.2549443	0.4734466	0.4791113	0.4762621
PCA	**0.3860860**	**0.3878723**	**0.3869771**	**0.5546999**	**0.5616488**	**0.5581528**
RP	0.1137931	0.1105053	0.1121251	0.1156499	0.1123152	0.1139581

Table 6.5: Shows the performance of involved topic modeling methods with different extracted topics t, t=20 and 50, (average value of recall, precision, and F-score).

TM Method	**Number of Topics**					
	20			**50**		
	R	P	F	R	P	F
LSA	0.2198939	0.2128799	0.2163301	0.2210345	0.2279532	0.2294835
LDA	0.3446734	0.3435585	0.3489088	0.2312177	0.2174433	0.2336483
NMF	0.5918747	0.5977849	0.5948151	0.6915826	0.6952324	0.6934027
PCA	**0.6339618**	**0.6392421**	**0.6365910**	**0.7044610**	**0.7086668**	**0.7065576**
RP	0.1132626	0.1106185	0.1119249	0.1084881	0.1052548	0.1068470

Table 6.6: Shows the performance of involved topic modeling methods with a different number of features f, f=10,100,1000, and 10000, (average value of recall, precision, and F-score).

TM Method	Number of Features			
	10	100	1000	10000
	F-Score	F-Score	F-Score	F-Score
LSA	0.108238987	0.177539633	0.196284973	0.187135878
LDA	0.118222579	**0.313004427**	**0.596767795**	**0.616768865**
NMF	**0.124100619**	0.246607097	0.384831984	0.478534632
PCA	0.118742505	0.273855576	0.459150019	0.553060479
RP	0.123841731	0.101052719	0.126599635	0.114772128

We observed that each topic modeling method we used had its strengths and weaknesses. During our evaluation, the results of all the methods performed similarly. Briefly, by comparing the outcomes of the extracted topics, PCA produced the highest term-topic probability; the NMF, LDA, and LSA models provided similar performance; PR statistical scores were the worst compared to the other methods. The probabilities ranged from 0 to 1 in all evaluated topic modeling methods; however, it provided a selection of non-meaningful words, like domain-specific stop words that are not suitable for further processing.

We also noticed that the LDA method provided the best learned descriptive topics compared to the other methods. Aside from some methods that failed to create topics that aggregated related words, like the LSA topic modeling method which usually performs best at creating a compact semantic illustration of words in a corpus. In addition, in the above Tables, the PCA and RP methods had the best and worst statistical measure results, respectively, when compared to other topic modeling with similar performance results. However, the PCA and RP methods distributed random topics, which made it hard to obtain the main topics from them.

Moreover, the LDA and NMF methods produced higher-quality and more coherent topics than the other methods in our evaluated Facebook conversation dataset, but the LDA method was more flexible and provided more meaningful and logical extracted topics, especially with fewer numbers of topics that matched our final aim of determining a topic modeling method that could understand the online user-generated content. Also, when comparing the LDA and NMF methods based on their runtime, LDA was slower and it would be a better choice to apply NMF specifically in a real-time system. However, if runtime is not a constraint, LDA outperforms the NMF method. The NMF and LDA have similar performances, but LDA is more consistent. Figure 6.7 presents the F-score average results with different numbers of features.

We ran all the topic methods by including several feature numbers, as well as by calculating the average of the recall, precision, and F-scores. As a result, the LDA method outperformed other topic modeling methods with the most features, while the RP model received the lowest F-score in most runs of our experiments. The following graphs present the average results of the F-score with a different number of features f on the Newsgroup data. Aside from the topic modeling method comparison, the graphs show that a higher F-score was obtained with the LDA model.

In addition, over the Facebook conversation data, the LDA method defined the best and clearest meaning compared to other examined topic modeling methods. Figure 6.7 shows the F-score average results with different numbers of features f= 10,100,1000,10000 (20-Newsgroup dataset). Furthermore, we measured the topic coherence score and we observed that extracting fewer numbers of keywords led to a high coherence score in the LDA and NMF topic modeling methods. As a result, obtaining fewer keywords can help define the topic in less time. This is useful for our future development of a real-time SRS that aims to analyze the user's online conversation and deliver a suitable task such as advertisement.

Based on our experiments, we decided to focus on LDA methods as an approach to analyze short social textual data. The LDA topic modeling is a widely used method in real-time SRSs and one of the most classical state-of-the-art unsupervised probabilistic TMs. They can be found in various applications in diverse fields such as text mining, computer vision, social network analysis, and bioinformatics (Vulić et al. 2015), and (Liu et al 2016).

6.3.7 The Topic Modeling Methods Third Evaluation:

In the LDA model, the corpus is organized as a random mixture of latent topics. Each topic refers to a word distribution, where each topic is a probability distribution over words. A major advantage of using the LDA model is that topics can be inferred from a given collection without input from any prior knowledge. On the other hand, it may not be able to capture important structures or extract meaningful, interpretable, and coherent topics. Thus, we tried to extend the LDA model to address this issue and to uncover hidden semantic themes more effectively. Indeed, it is a challenging task to accurately interpret the real meaning of the conversation's topics by only using the word distributions extracted from the corpus.

Furthermore, some researchers have emphasized that the LDA model performed well on texts that include at least a few hundred words like in (Jonsson and Stolee (2016); Qiang et al., 2019). Consequently, we applied both BoW and TF-IDF algorithms as a pre-filtering stage before the LDA to overcome this issue. The BoW method handles a document as a collection of terms and ignores the sequencing of words in a document and the language structures such as grammar.

(Pascual, 2019) stated that the LDA model has three hyperparameters: the number of topics (t), which is usually defined by the analyser ahead of time, the alpha hyperparameter (α), which disturbs the number of topics to each document; and finally the beta hyperparameter (β), which disturbs the number of words in each topic. A higher α means that each text will be represented by more topics and a lower α has the opposite effect. Conversely, a higher β means each topic is represented by more words, while a lower β means we would get less "overlap" between topics.

Definitely, adding both algorithms before applying the LDA model leads to better results and helps to understand the conversation better, besides obtaining higher-quality results that improve our ChatWithRec SRS. However, the LDA model is still considered as a central model and has revolutionized the TM field. As well, it has proven to be a standard and effective technique over the years to extract the hidden topics from a vast textual data (Mazarura J, 2014). Moreover, we trained the LDA topic model again on the Wikimedia articles and 20-Facebook conversation datasets. Table 6.7 shows a sample of our Wikimedia dataset results. In addition, we tried to extract different numbers of topics in our final evaluation to improve our ChatWithRec system's accuracy and usability.

Table 6.7: Shows sample results from Wikimedia articles dataset:

File Name	# of Topics-5 keywords in each	Extracted Keywords
LDAmodel10_100	10	[['"thank" ', '"want" ', '"love" ', '"Facebook" ', '"next"'], ['"pm" ', '"see" ', '"new" ', '"conversation" ', '"family"'], ['"good" ', '"travel" ', '"know" ', '"many" ', '"chat"'], ['"friend" ', '"work" ', '"need" ', '"meet" ', '"fine"'], ['"long" ', '"think" ', '"kind" ', '"mutual" ', '"first"'], ['"lot"

		', '"Saudi" ', '"place" ', '"problem" ', '"back"'], ['"university" ', '"program" ', '"plan" ', '"app" ', '"cook"'], ['"time" ', '"would" ', '"night" ', '"system" ', '"take"'], ['"pasta" ', '"end" ', '"month" ', '"science" ', '"jersey"'], ['"send" ', '"go" ', '"study" ', '"week" ', '"nice"']]

We tested the modified LDA models over our 20-Facebook conversation data and measured both coherence and F-score, as shown in Figures 6.8 and 6.9. To define the best number of the extracted topics in the short text, we considered the topics shown in Table 6.8. The coherence score shows how well a topic represents the corpus from which it was modelled. Its applies the word co-occurrences in documents to train the model. The co-occurrence matrix includes two document terms in rows (R) and in columns (C), then the matrix shows the number of times each R term appears in the same context as each C term. From our examination, we observed that extracting fewer keywords led to a high Coherence score.

Table 6.8: Shows a sample results from the 20-Facebook dataset.

Number of topics	Extracted keywords
2	[[“travel”], [“time”]]
5	[[“good”], [“travel], [“new”], [“lot”], [“time”]]
10	[[“time”], [“many], [“good”], [“plan], [“new], [[“square”], [“family], [“decide”], [“travel”], [“back]]
15	[“family”], [“ place”], [“ good”], [“ love”], [“ next”], [“see”], [“lot”], [“many”], [“travel”], [“park”], [“time”], [“buy”], [“spend”], [“see”], [“square”]]
20	[[“central”], [“buy”], [“decide”], [“plan”], [“park”], [[“time”], [“family], [“love”], [“week”], [“see”], [“next”], [“many”], [“good”], [“buy], [“new], [[“plan”], [“family], [“square”], [“good”], [“jersey”]]

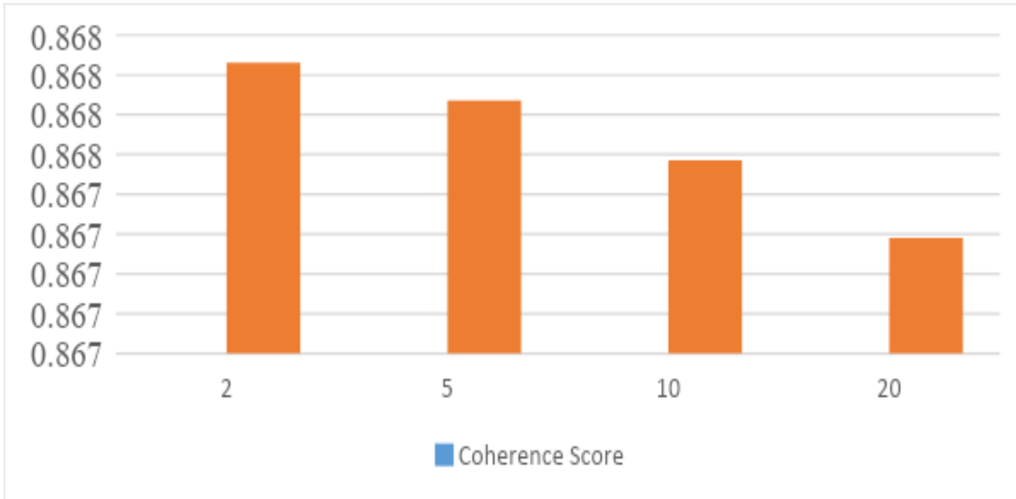


Figure 6.8: The result of Coherence score for in the 20-Facebook dataset.

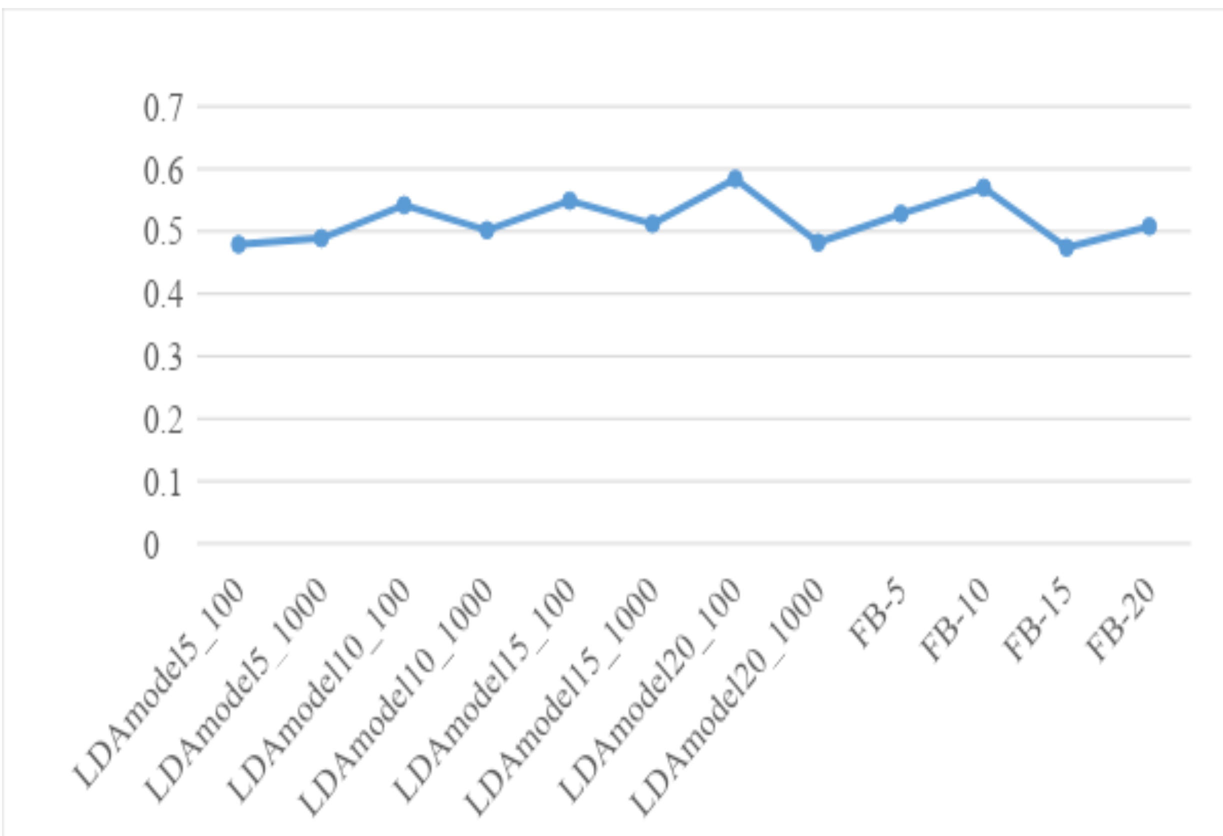


Figure 6.9: The result of F-score in the 20-Facebook and Wikimedia articles datasets.

As we can see from above Figures, the measurement of coherence score is in a range of 0 – 1 and "if a measuring value is closer to 1, that means that the model has correctly identified words that co-occur frequently in documents as a topic, but this cannot guarantee that they make semantic sense or that they are interpretable by a human; it simply means that the topics represent the data known to be in the corpus" (Mimno D, 2011). Furthermore, obtaining fewer keywords can be helpful to define the topic in less time, which is useful in our developing our SRS. As a result, we decided to use the modified LDA topic modeling method in our proposed ChatWithRec SRS, since it provides good results that match our system's intention. Adding both algorithms before applying the LDA model leads to better results. It helps to understand the conversation better and obtain higher-quality results that improve our SRS.

6.3.8 Summary of the Initial Evaluation

TM has been applied to numerous areas of study such as Information Retrieval, computational linguistics, and NLP. It has been applied effectively to clustering, querying, and retrieval tasks for data sources such as text, images, video, and genetics. This work delved into a detailed description of some significant applications, methods, and tools of TM; it focused on understanding the status of TM in the digital era. In our evaluation, we used several SM textual datasets. The performances achieved by TM methods were compared using the most important and common standard metrics in similar studies, namely recall, precision, F-score, and coherence. We also defined which methods can deliver maximum well-organized and meaningful topics.

As a result, we found that all of the included topic modeling methods we used had much in common. For example, they all included transforming text corpora into term-document frequency matrices, using the TF-IDF model as a pre-filtering model, and producing topic content weights for each document, among other processes. Despite these similarities, the two TM methods that generated the most valuable outputs with diverse ranges and meanings were the LDA and NMF TM methods. Finally, we discussed the evaluations performed to verify the correctness of the topic extracted from the user's online generated content by using the TM method. We decided to use the modified LDA method in our proposed ChatWithRec SRS since it provided good results that match our system's intention.

6.4 The ChatWithRec SRS final Evaluation

In this section, we focus on the user cold-start problem that occurs when a user registers to the system. The system has no information about his/her personal information, history, previous interactions, and preferences. Consequently, the system fails to recommend anything to him/her because it couldn't form a user profile to match significant items and like-minded users. The new user cold-start problem exists in both CB and CF filtering approaches. To build the user profile and produce suitable results, there should be enough user feedback, which generally takes the form of the item's ratings (Lika, Kolomvatsos, 2014).

An example is (Chughtai. M, 2014), who merged CF with a CB of learning content to solve the user cold-start problem in the e-learning domain. We will propose a possible solution by merging CF with CB filtering recommenders as a hybrid method that is good at predict the online behavior of the users in SM platforms and provides accurate recommendations. We only consider the user online generated content as a source of data in the recommendation calculation process, where there is no need to check the user's preferences (detect his earlier activities like item's ratings, perceive his previous purchase history, apply some similarity method, social graph, or other offline types of actions). Hence, to offer satisfactory services to the users, SRSs should effectively recommend new users/items when they enter the system for the first time.

We focus on verifying the delivered recommendations after analyzing the user's online text data, in addition to verifying the effectiveness of our proposed ChatWithRec real-time SRS. Firstly, we are going to describe the setup environment of our evaluations, including the diverse kinds of datasets we used, and then describe the pre-processing procedure we applied for all included textual datasets as an input source. In our initial ChatWithRec evaluation procedure, we examined many TM methods to determine which method was appropriate to integrate into our system; we concluded that this was the modified LDA topic modeling method (based on BOWs and TF-IDF). We used two advertisement databases as real-time output recommendation modes. We conducted experiments for each of the advertisement recommendation modes to evaluate the correctness and effectiveness of ChatWithRec SRS.

6.4.1 Evaluation Methods

Many SRSs researcher are progressively attempting not only to evaluate the accuracy of recommendation methods and algorithms but also to define the user satisfaction with recommendations created by them. In recent times, many evaluation methods are based on several techniques. Some examples of these techniques include the online approach, the offline approach, case-studies, user surveys, and questionnaires where users answer questions evaluating their enjoyment and satisfaction among numerous other factors. Consequently, in our evaluation process, we measured the predictive power of our proposed ChatWithRec system by using both offline and online evaluation approaches.

Our evaluation is appropriate for obtaining the performance and accuracy of two topic modeling methods that are going to integrate into the ChatWithRec system to analyze the user online generated-data. In Chapter 6, we used the offline evaluation approach to compare several topic modeling methods on various short textual datasets (Hu and Ogihara, 2011).

In our online evaluation, we performed a case-study evaluation that recruits a group of users to assess an SRS, which allowed us to collect qualitative data and to define the user's within-subjects interaction with the system. It was designed to estimate the recommendation performance collected via surveys or questionnaires that included many questions to obtain the user's satisfaction and behavior toward the proposed system. This includes information such as whether they liked the system and have received interesting recommendations (Shani and Gunawardana, 2011).

6.4.2 Evaluation Metrics

In general, the goal of evaluating ChatWithRec SRS is to define whether it detects the topic that the user is talking about in real-time (same conversational session) and whether it recommends interesting items that are related to the user's current need which he might view, buy or listen to (overcome the user cold-start problem). In this subsection, coherence scores are used to evaluate the performance and accuracy of ChatWithRec SRS, which generates satisfactory results compared to other performance evaluator metrics in this area of SRSs like error metrics.

The coherence score is described as how interpretable a topic is based on the degree of significance between the words within the topic itself. We calculated the coherence score with different topic number t, t= 2, 5, and 10 extracted topics to define the system's effectiveness. Furthermore, we investigated user satisfaction by using a survey questionnaire that measured the user's self-reported satisfaction.

Our developed survey questionnaires contained many dimensions, that aimed to measure user fulfillment over three principal elements: perceived personalization, perceived enjoyment, and perceived ease-of-use. We recruited 27 participants in the usability survey (11 males and 16 females; ages: 18-44 years) as shown in Table 6.9. In this evaluation sector, we only considered 10 participants who had completed all three evaluation sessions. We believe that our number of participants was limited by the rapid growth of **COVID-19,** which emerged in Wuhan, China, in November 2019, then has grew into a global threat and was eventually declared a global pandemic (Q. Li, 2020) and (S. Kumagai, 2020). However, the problem in usability test expect to be 85% detected with 4 or 5 participants as stated in (Nielsen, 1994; Nielsen, 2012).

Table 6.9: Participated users' information (Gender, Occupation, and Type of smartphone).

User	Gender	Type of Smartphone	Occupation
1	Male	iPhone 7	Graduate Student
2	Male	iPhone 11 Pro	Undergraduate Student
3	Female	iPhone 8	Graduate Student
4	Male	iPad	Graduate Student
5	Female	iPhone 7 Plus	Graduate Student
6	Female	iPhone 6 S	Employer
7	Female	iPhone 11 Pro	Undergraduate Student
8	Male	iPhone 10	Graduate Student
9	Male	iPhone 8	Employer
10	Female	iPhone 6 S	Graduate Student

We also explained to participants how to use the ChatWithRec app in the tutorial session and explained how they could embed the recommendation of the ad in their chat session. All participants were asked to use the app three times. Each time, all participants were asked to conduct a conversation according to given tasks which were the same for all users (making a plan for a trip, a plan for dinner, and booking a flight\hotel). Furthermore, our experimental datasets were gathered based on connections between real humans to guarantee that the corpus could completely simulate real human conversation behavior. Table 6.10 includes the survey questionnaires.

Table 6.10: The questions in the survey questionnaires about user satisfaction.

Subjective Factors		**Questions Q**		**How users answered the Questions**
Perceived	Enjoyment	Q.1	Is recommended ads enjoyable to brows?	On a 5-point likert scale from strongly disagree to strongly agree
	Personalization or Usefulness	Q.2	Are recommended ads personalized for me?	
		Q.3	Is the ChatWithRec interface capable of helping you effectively find ads that you are interested about?	
	Ease of Use	Q.4	Is the ChatWithRec app easy to modify and control?	
		Q.5	Do you find ChatWithRec app easy to use?	

6.4.3 Evaluation Procedure

The experiment in this section was carried out on two datasets (Wikimedia articles and 20-Facebook datasets) to evaluate the effectiveness and user satisfaction from using ChatWithRec

SRS. We recruited 27 participants and all participants were asked to use the app 3 times. Each time, all participants were asked to perform 3 topics tasks, which were the same for all users. The time difference between the first and second sessions was one hour. Between the second and the third sessions, there was a one-week interval. Our data was a collection of user-generated dialogues concerning various topics like travel, ordering food, and shopping. After collecting the data, we applied data pre-processing which is a text cleaning method to prepare it before sending it to the LDA model.

After the preprocessing step, our server applied the BoW and TF-IDF algorithms as a pre-filtering stage. We used the LDA model afterward to examine and distinguish the topic discussed by users. The server then sent extracted topics to our advertisement database to find related ads that matched a user's real requirement, then recommended them to him/her during his/her chat flow. As part of the interaction, the users performed several actions, such as chatting, browsing the ads, and/or requesting that specific or repeated ads stop being shown. Figure 6.10 presents a snapshot of the ChatWithRec app environment.

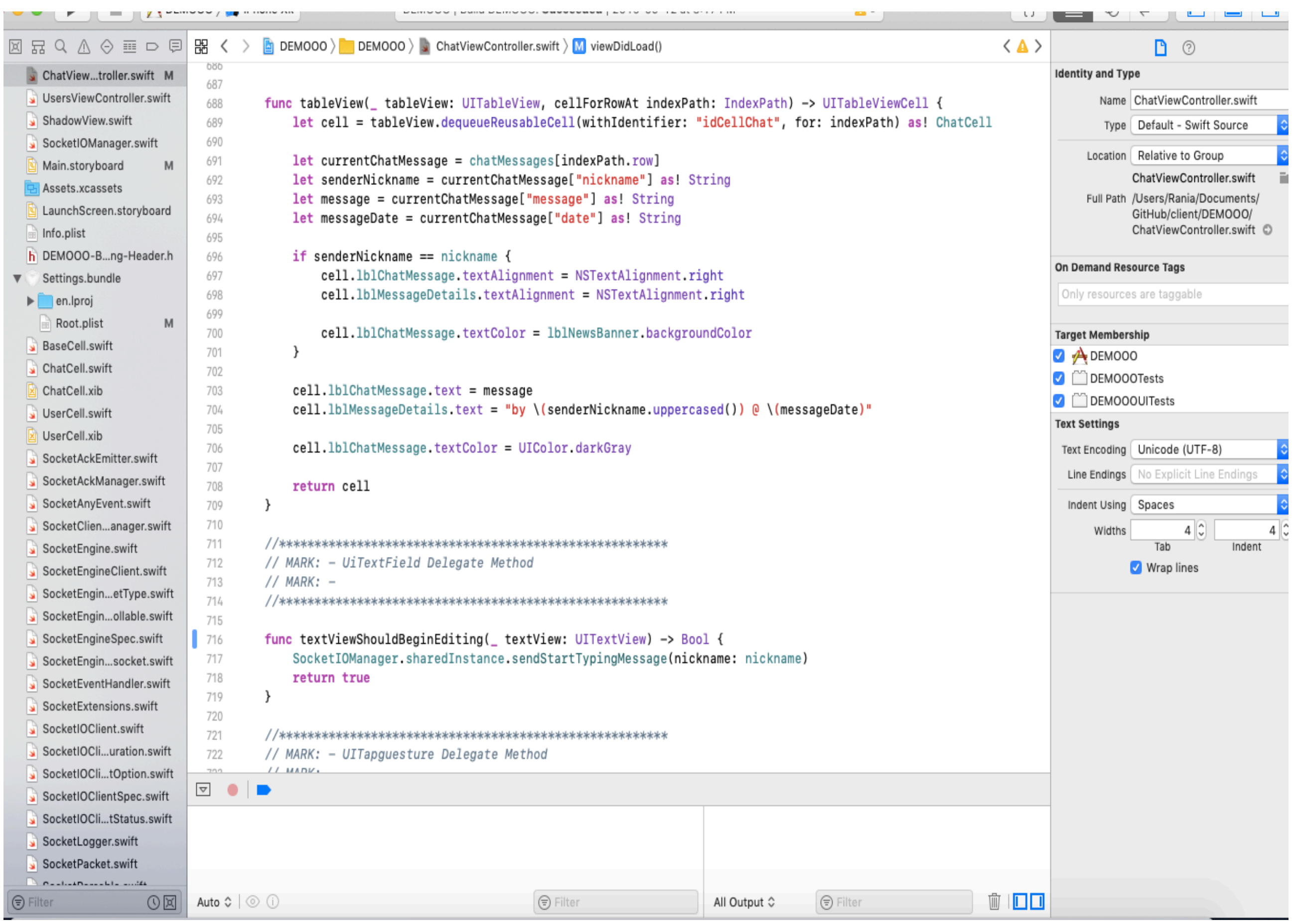


Figure 6.10: A snapshot from the ChatWithRec environment.

We tested our ChatWithRec SRS via many scenarios to check whether our application ran smoothly and to evaluate our system's quality and usability; for example, one scenario involved shopping and included an exchange between a real device user and the system simulator. Both users can log into the app by providing a username after subscribing to our system. Then they can enter the online user list and easily start a conversation with another online user. Hence, there is no need for user registration at this stage to receive personalized recommendations because we only considered analyzing the user's conversation dynamically to deliver conversation-related content in order to solve the cold-start problem. The conversation session was between 2-7 minutes in duration and the system was able to detect the topic and suggest a related advertisement within 3 minutes or less. Another scenario dealt with going out to eat and was between two real devices, as shown in Figure 6.11. This scenario lasted approximately 15 minutes.

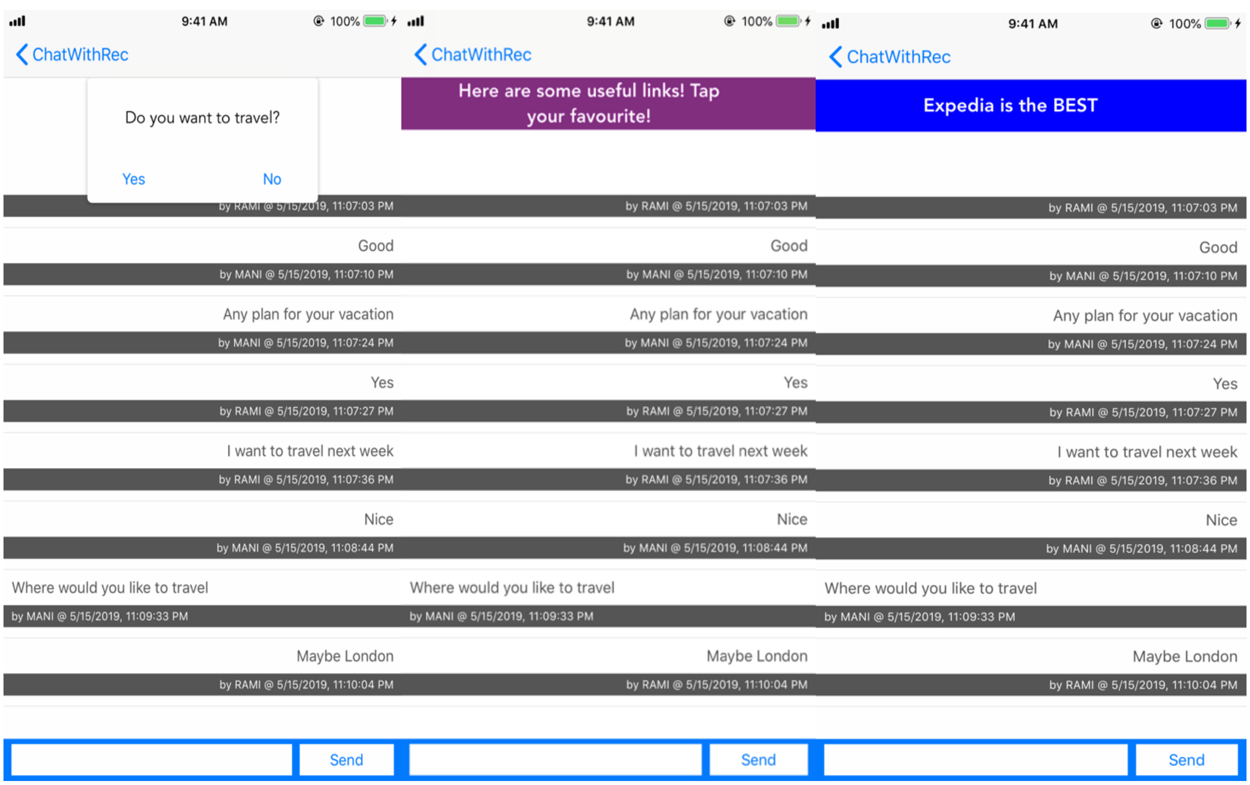


Figure 6.11: A snapshot from ChatWithRec experiment.

We received many advertisement recommendations; one of them was about food application ads. At the beginning, we showed the ad as interstitial ads (full screen). We noticed that it bothered users, since it appeared as a full-screen advertisement. Consequently, we decided to display the recommended ads as a banner advertisements and not as interstitial ads. This was done to ensure the ads were not an annoyance and to maintain the user's conversation flow.

6.4.4 Evaluation Results

In our experiment, we extracted the user's conversation topics manually to make sure our proposed recommendation system could understand the conversation as well as extract the right context topic from it. We should mention that we displayed advertisement recommendation links after sending a popup message. These messages were sent to the subscribed user regarding the ad topic the server predicted from their conversation flow that our system detected to make sure it matched their interests. Hence, this step enhanced the users' interaction, since they felt that they could control the system output actions. Again, the goal of these system was not only to improve the accuracy of the recommendations but also to provide an effective user-recommender interaction. As a result, subscribed users received advertisement recommendations that were most interesting to them and met their needs based on their preferences, as shown in Figure 6.12.

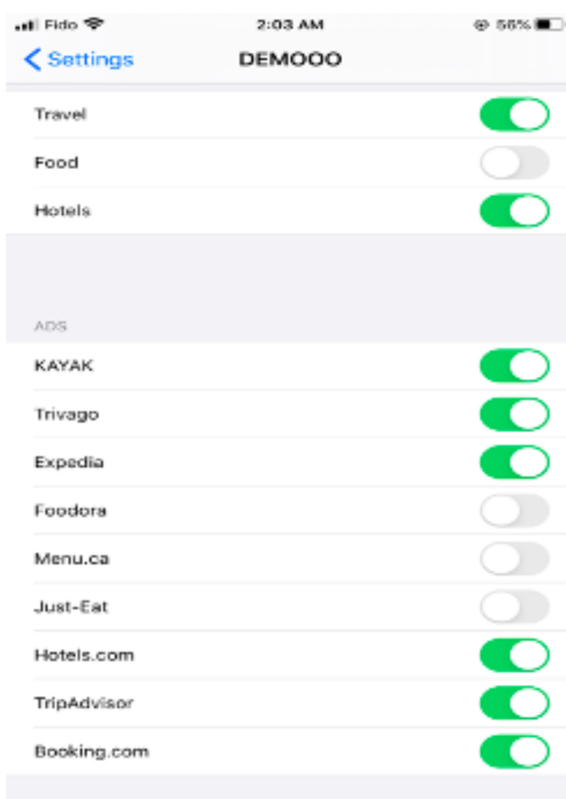


Figure 6.12: A screenshot of the user preference page.

Since we relied on the user's online generated content, there was no need to check the user's friend preferences, item ratings or other features to send the advertisement's notification to the active interested user.

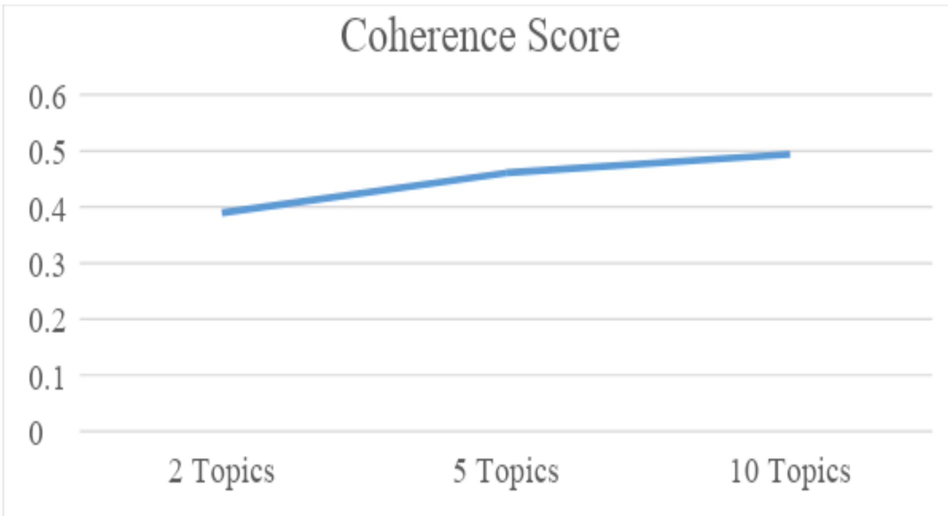


Figure 6.13: The Coherence score results.

Figures 6.13 presents the results of the Coherence score in one sample from the 20-Facebook conversational dataset with different extracted topics. The topic coherence measures used in this work aim to evaluate the quality of topics from a human-like perspective. From the above Figure, it is clear that extracting more topics led to an increase in the coherence score. Lastly, we obtained satisfactory and reasonable results from our survey questionnaire, as shown in Figure 6.14. However, we only included 10 real users in our online evaluation, which makes it difficult to generate final results, so adding more users is essential to validate and enhance ChatWithRec SRS.

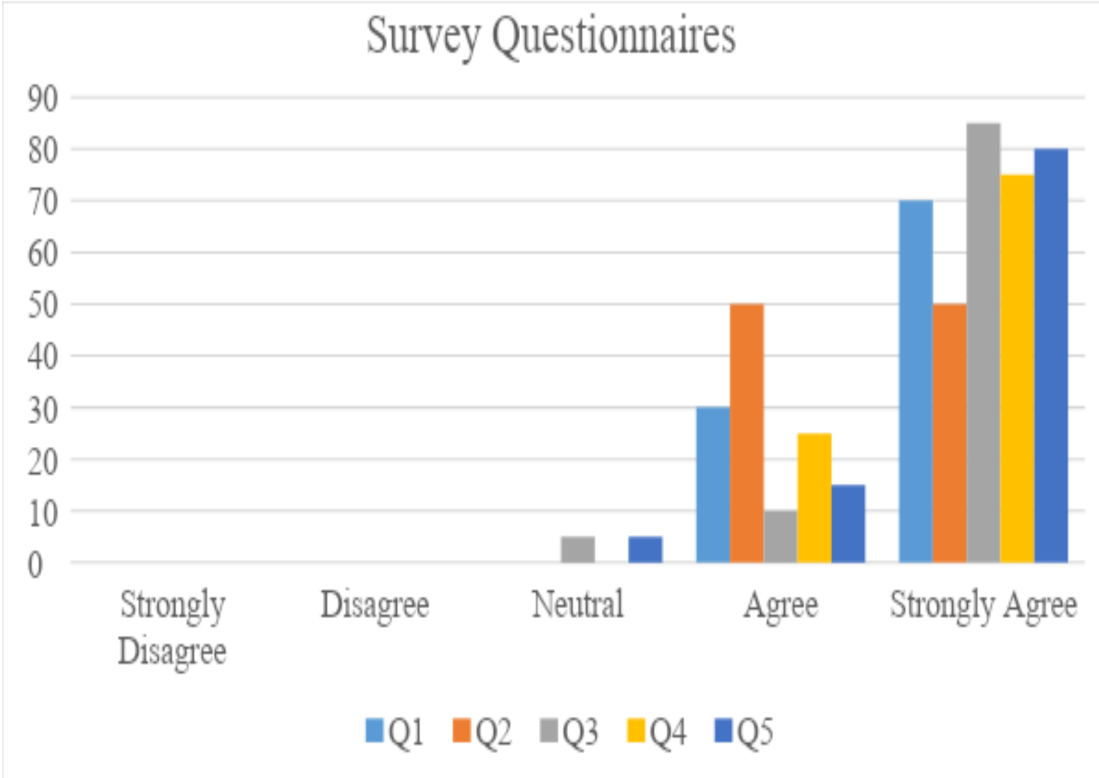


Figure 6.14: The survey questionnaire results.

Figures 6.15, 6.16, and 6.17 present our results from three experiment sessions that were related to determining the effectiveness of and user satisfaction from using ChatWithRec. Furthermore, we observe that users tended to check at least one recommendation when they were either neutral or agreed that the recommendation list contained ads they thought they would enjoy browsing and met their needs. More users agreed that the more the recommendation lists were personalized to them, the more likely they would be to take at least a recommendation from the list. Less than 30% of users who did not enjoy the presented recommendations took at least one recommendation. On the other hand, more than 85% of users who answered that they enjoyed recommendations (either Agree or Strongly Agree) did take at least one recommendation. Users who enjoyed or did not enjoy the recommendations were those who were more likely to try out many suggested ads. All subscribed users received advertisement recommendations that were related to their conversation sessions. We also acquired a high F-score with smaller numbers of extracted topic.

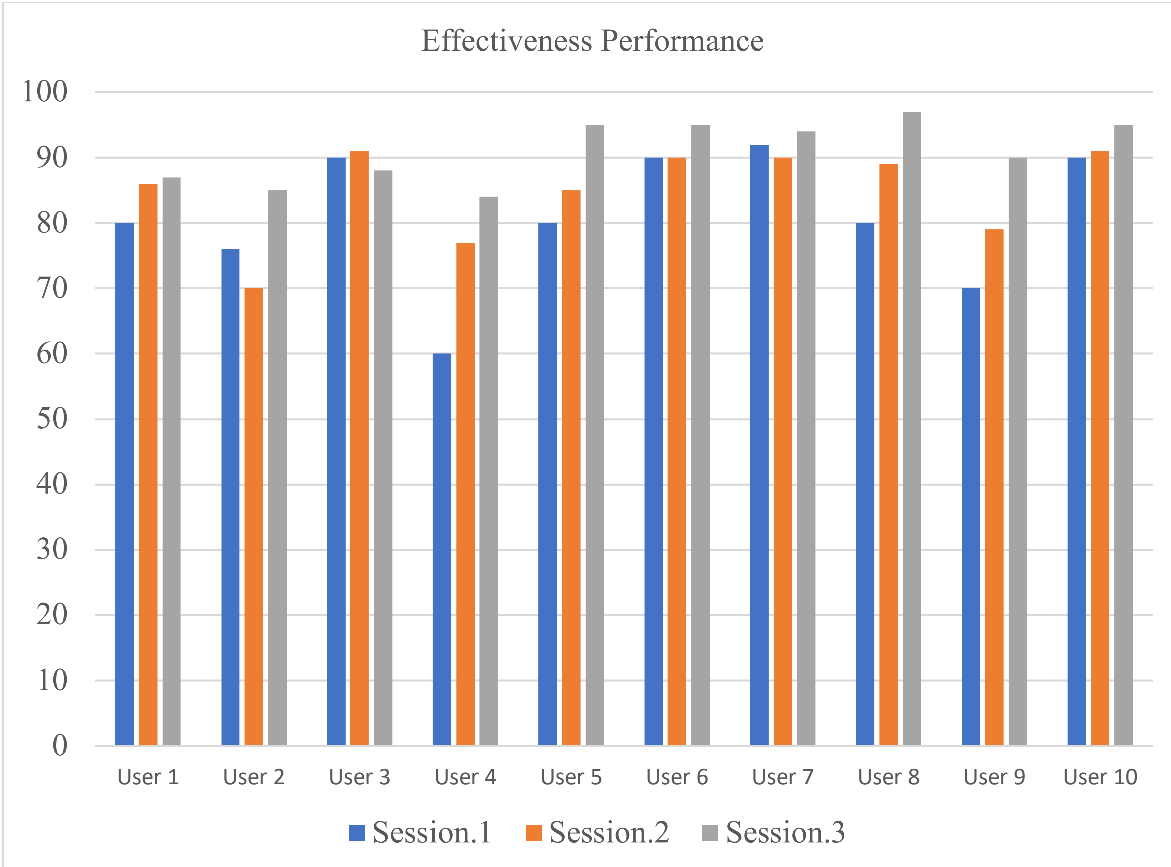


Figure 6.15: The user's effectiveness Performance during 3 experiment sessions.

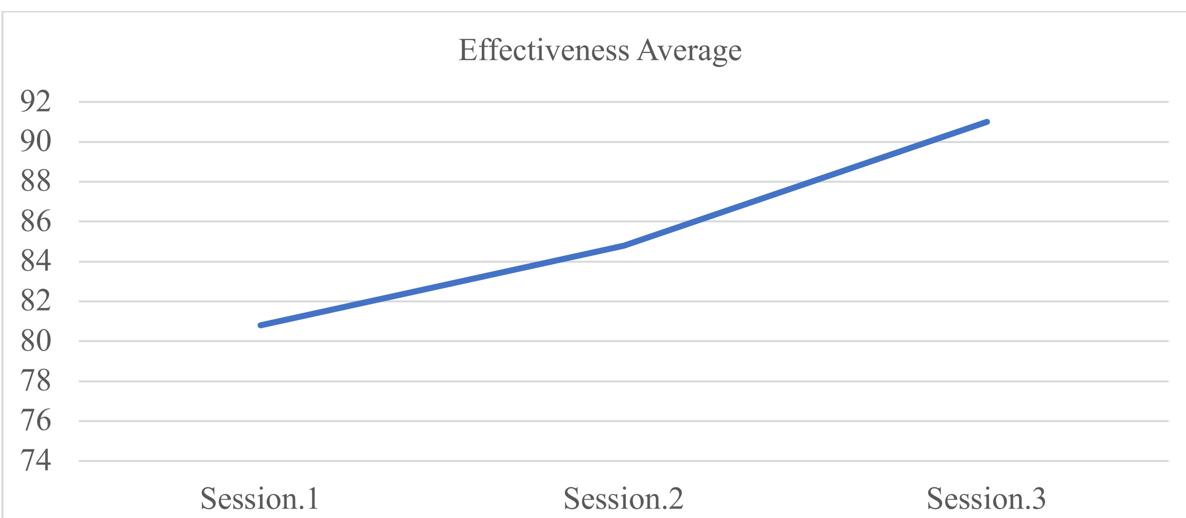


Figure 6.16: The effectiveness average in all included experiment sessions.

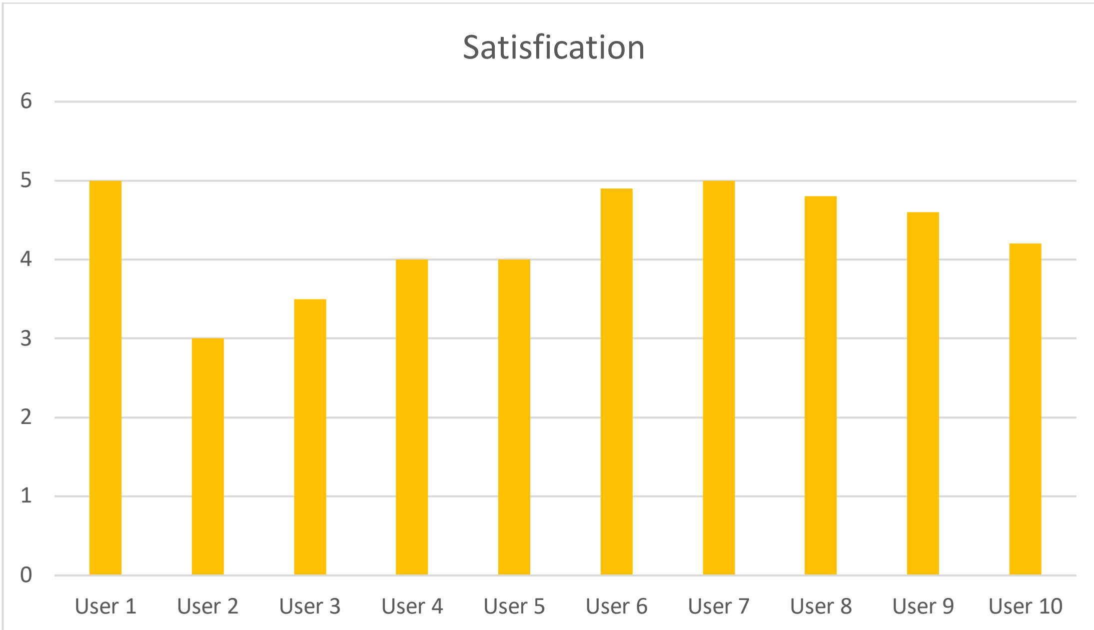


Figure 6.17: The average satisfaction score.

From the above Figures, it's clear that the user's performance increased over time, indeed, many participated users enjoyed talking more time in their conversation to experience more advertisement recommendations from different e-commerce providers. Furthermore, after each experiment, participants answered a survey questionnaire on their real-time social recommendations experience and the effect of using ads on mobile chats. we additionally employed an information systems-oriented evaluation based on the precision metric, as in Equation.9 and Table 6.11.

Precision= (total relevant ads) \ (total recommended ads) Eq.9

Table 6.11: The information systems-oriented evaluation results.

User	No. Comments	Total No. words	No. ads clicks	Relevant ads	Recommended ads	Precision
1	16	806	5	8	11	73%

2	20	911	5	10	14	71%
3	59	2408	8	15	15	100%
4	25	1648	6	9	11	82%
5	22	171	2	7	8	87%
6	45	1983	7	15	16	94%
7	50	2011	4	6	6	100%
8	17	408	2	4	5	80%
9	46	3090	7	9	10	90%
10	12	238	3	5	6	84%
Average						86.01

6.4.5 Summary of the Final Evaluation

We aimed to extract the real meaning from user conversation dynamically, rather than just to extract a list of keywords. Subsequently, adding more topic modeling algorithms to filter our text data is necessary to make ChatWithRec SRS more efficient and valuable to use. Without a doubt, effective SRS should consider NLP approaches like sentiment, intent, and entity recognizers, besides recommendation algorithms that help to define and investigate many important tasks from online social networks. Many existing SRSs involves numerous techniques that are based on many methods and different data principles, such as users' content preference, demographics, location preference, social closeness and item details.

We presented the general evaluation of our proposed ChatWithRec SRS that we aim to use to overcome some recommendation system limitations such as cold-start and data sparsity problems. The results indicate that we solved the cold-start problem since there was no need to involve user chat history, preferences, and profile data to define suitable advertisements for users. We get the benefits from analyzing a user's contextual conversation dynamically to define and detect their main conversation topic to take action and send them related tasks which in our thesis was advertisements (will add more task-related action in future work). During our development, we adopted the LDA model in Gensim format by using BoW and TF-IDF methods; both methods lead to similar results. We also involved AdMob and our adjusted databases as a task-related output action to satisfy users during their conversation flow. In addition, during our evaluation we faced some limitations in our evaluation process that we tried to overcome by enhancing our system's functionality, which is known as a feedback loop procedure:

- The ChatWithRec system used the AdMob advertisement database when we could not detect the conversation's topic in a short time (less than 3 minutes). However, AdMob showed random advertisements in some cases, since it considered the user's location more than detected topics from our system. These ads were not satisfactory, since they didn't reflect any of the user's conversational content or topics of interest. Obviously, we could not control AdMob's manufacturing action side after we sent the extracted topic from our server side. To solve this issue, we increased the topics and advertisement categories to cover many recommendation subjects. This solution increased the user's satisfaction regarding our provided recommendations. Moreover, we considered the advertisement presentation by considering several aspects, such as the ad's size and other technical features.

Table 6.12 presents a comparison between our proposed ChatWithRec and many existing similar SRS architectures. As we can observe from the table, our system outperformed related work by providing a real-time, mobile, conversational system that can provide more personalized recommendations immediately to interested users based on their online generated-content as the main source of data besides the user's location and preferences.

Table 6.12: A comparison of our proposed ChatWithRec SRS against some similar existing SRS architectures.

Similar Works	Type of recommendation	Real-Time	Sources for CSP	Methods used	Evaluation Metrics	Remarks
(R.Preethi, 2021)	Recommender's products catalog coming from e-commerce sites.	NO	Amazon and Yelp datasets	deep neural network-based LDA with DNN Sentimental Analysis	Time Accuracy	Recommender's products to tiny blogging customers who do not need history of buying records.
(V. R. Revathy ,2019)	Item Recommendation like movie.	NO	User registration, social network data, contextual data, and ratings	CP, clustering, implicit feed back	Not Identified	Considers data inputs from all the additional sources and collects additional data during user registration.
(Yuchen and Zhang, 2016)	Advertisement Recommendation.	YES	User's preferences	Retrieval algorithm, Top-k aggregation algorithm, Safe region method	Read/write ratio.	Users may get the same set of top-k advertisements if the dynamic context of users stays the same or changes a little.

(Lin et al., 2013)	iTunes App to Twitter users.	YES	Social network information	LDA	Precision Recall	Social media site ffollower's information is the only primary source
Arain, Q. , 2017)	Travel Recommendation.	NO	Social context information Geo-location information	Probabilistic Bayesian Method, Kernel density estimation, Symmetric Kullback–Leibler divergence	Mean average precision	Dependent on historical location knowledge
Proposed work **ChatWithRec SRS**	Real-time Advertisement recommendation	YES	Online UGC User Preferences location	Modified LDA	F -score Recall Precision Coherence score	Based on the user's online generated textual data to provide more personalized recommendation in real-time manner

Chapter 7 Conclusion and Future Works

Today, people tend to rely increasingly on the Internet in their social and commercial activities. Indeed, the Internet assists in increasing the demand for the development of business applications and services that can provide better shopping experiences and commercial activities for customers around the world. The Internet is full of information and knowledge sources, which might confuse readers and make them spend additional time and effort trying to find relevant information about specific topics or objects. Recommendation systems are considered to be an essential method to solve these issues. It is the most popular and productive way to guide users to the information on the web that meets their needs and preferences.

Researchers in both academic and industrial fields have demonstrated the importance of recommender systems in e-commerce in both websites and applications. They have also developed numerous methods and techniques in recommendation systems to help users receive recommendations that are relevant to them socially and personally at the appropriate time. Researchers have also created many techniques to collect users' information, like their data, past purchase history, and behavior to provide new and more appropriate recommendation lists that meet the users' interests and to enhance commercial activities in general.

The research of this thesis follows the Design Science Research method, which validates its process by following the six guidelines (Peffers et al., 2007). We studied other relevant research in NLP and social recommendation systems and real-time architecture development from educational and industrial partners in order to develop our ChatWithRec. Our system aims to work as a third-party system that tracks the user's generated content in a real-time manner, unlike the cookies approach that might be blocked on some users' machines in response to security or privacy concerns, thereby causing problems in delivering accurate recommendation to online users.

We aimed to improve SRS accuracy, usability, and performance by proposing a real-time SRS that is integrated into most existing systems by applying multiple approaches such as NLP and semantic rich search solutions. In fact, by detecting the users' conversations we can define more information about them, for example, their intentions in real-time. This motivated us to build our system. In addition, in our proposed ChatWithRec, we explored many NLP algorithms, such

as NMF, LDA, LSA and other unsupervised topic modeling methods, In the end, we adopted a modified LDA topic method that used both BoW and TF-IDF algorithms as a pre-steps. The integrated LDA model was used to detect topics related to users' conversation sessions.
We also involved two advertisement approaches in our implementation: the AdMob database, which sometimes showed random advertisements in some test scenarios; and a modified database which included a specific number of linked websites as recommendations. Based on the AdMob issues, we decided to reduce our targeted areas and filter our recommendations by considering only certain fields, namely, food and travel subjects including booking hotels and flight adverts. In our future work, we will increase our areas to have more topics, such as shopping.

Moreover, users are often not satisfied with the initial recommendations because some of those models will not elicit the users' preferences at the beginning of the interaction and there remains a need to study the conversation context in order to provide better-related tasks. In our current implementation, we attempted to solve the user cold-start problem by only including online user conversations.

Our proposed system extracts and analyses conversational context to identify topics dynamically while users are interacting with each other. This creates an action based on a defined topic, but before sending a recommendation to a user, we ask first if he/she is interested in the detected topic. If the answer is, " yes", then he/she can receive several related links to give the user the opportunity to choose what he/she prefers. The experimental results of our proposed system offer promising gains in enhancing the accuracy of the prediction and defining the most suitable recommendation for online social network users.

In the following section, we explain how we addressed and answered the research questions that were derived from the central research question:

How to derive a real-time social recommendation from the user's online social network platforms and apps?

- ***What are the primary features and criteria in designing the main architecture of an efficient real-time social recommendation system?***

In the beginning, we considered four basic features in our proposed ChatWithRec that are fundamental in many existing recommendation systems to make it more effective. These features are: personal, mobile-based, real-time, and text-based SRS. We believe that offering an RS that can understand the real context of users' online conversations is essential to recommending interesting items at the right time. Delivering dynamic recommendations to users that match their current interests and requirements from anywhere and at any time is important in order to increase the user's satisfaction. Moreover, the ChatWithRec architecture is designed based on an current literature review of other existing SRS architectures that applied different intelligent server and mobile procedures.

We tried to provide an effective real-time social and mobile-based RS that users can rely on as an intelligent server. This approach is certainly useful and more efficient because it provides simple, straightforward, and light app software that can handle many simple functions, as well as a powerful server that can handle the primary system's functions. Chapters 2, 4, and 5 contain many details that are related to ChatWithRec's features, architecture, design, and implementation.

- ***How does the topic modeling method contribute to the social recommendation system to recommend related tasks based on the user's context in a real-time manner?***

Our developed SRS architecture includes several tasks: analyzing the user's conversation context, understanding and defining the conversation session topics, filtering them and matching them with appropriate advertisement recommendations, and sending the ads to the user's app through the internet to display them. During our research, we tested many topic modeling methods that were used in various related studies on many textual social datasets in order to determine which TM method was suitable to integrate into our ChatWithRec. Chapter 6 contains a massive experiment that related to the NLP topic, techniques, and topic modeling approaches.

As a result, we decided to adopt the LDA topic modeling method with some modification which added both BOW and TF-IDF algorithms as a pre-filtering stage before feeding the LDA model. Adding both algorithms before applying the LDA model led to better results and helped to understand the conversation better, This also generated higher-quality results, improving our ChatWithRec.

The proposed ChatWithRec system can detect and extract the user's online conversation to analyze it and define related topics; however, only subscribed users can receive related recommendations. A subscribed user can choose the kind of interesting topics about which he/she wants to receive ad recommendations from the app in a simple and fast mode. The user also receives a message during his/her conversation to check whether the system has detected his/her current requirements. Based on his/her action, he/she can display several ad recommendations or continue his/her conversation normally. Both online and offline evaluations are used to examine the usability of our ChatWithRec.

We applied numerous statistical metrics to evaluate and analyze the performance and accuracy of our system. It defined how quickly the recommendations worked and how precise they were. Also tested were the real user's acceptance and satisfaction through survey questionnaires, which aimed to improve our system's quality and performance besides measuring the system's speed and accuracy.

Moreover, we proposed a real-time SRS called ChatWithRec. It aims to improve the accuracy of users' advertisement recommendations as well as to solve the user cold-start problem. Cold-start refers to the difficulty in bootstrapping the RS for new users or new items, while the data sparsity problem occurs in recommendations because of the insufficient amount of the feedback data collected from the user and transactions. The novelty of ChatWithRec is a real understanding of the users' conversations by identifying keyword context dynamically to detect topics within the users' dialogues and then match them with fitting tasks like advertisements. We also aimed to provide an appropriate user-recommendation interaction to increase the accuracy of the recommendation dynamically.

This thesis provides a road map of research in the SRS domain since it employed the topic modeling methods within the social recommendation system setting. We proposed an application architecture for a real-time social recommendation system that aims to improve the performance and accuracy of recommendations. This architecture can work as a third-party real-time social recommendation system that could be integrated with many social network sites (e.g. Facebook and Twitter). Furthermore, our implemented ChatWithRec SRS was designed to improve the performance and accuracy of recommendations besides addressing the user cold-start problem in a real-time manner.

We also created a social network application (chat app) as a proof of concept to verify our developed social recommendation system architecture, which includes several tasks: analyzing the user's conversation context, understanding and defining the conversation session topics, filtering them and matching them with suitable advertisement recommendations, then sending these recommendations to the user's app through the internet. Hence, our proposed SRS architecture is not limited to the OSNs tasks; it might be developed to integrate with many existing industries, as well as relying on other online generated-content such as video and image.

Finally, there have been impacts on research productivity and training in the academic and industry domains because of COVID'19, emerged in Wuhan, China, in November 2019, then grew into a global threat and was eventually declared a global pandemic (Q. Li, 2020) and (S. Kumagai, 2020). Indeed, the environment surrounding researchers has changed significantly because of COVID-19 restrictions. Due to quarantine, we had to stop laboratory tests and only consider a small group of people who completed all three experimental sessions successfully. Undeniably, including more real users to evaluate our ChatWithRec system would increase the system's effectiveness. However, the experimental results of our proposed ChatWithRec architecture offer promising gains in enhancing the accuracy of the predictions and defining the most suitable recommendations for OSN users. This thesis also provides a road map of research in the SRS domain, since it employed the TM methods within the RS setting.

Future Direction

Our study is not without limitations. First, it focuses on one domain of recommendations - advertisements recommendations. Consequently, we hesitate to generalize our conclusions to diverse domains of the social recommender system. We are planning to perform the following:

- ❖ Explore further and add more topic modeling algorithms, such as Word2vec, BERT, and NLU to understand conversational context further. Hence, we already built some of the previous algorithms by using the Python environment but we have not tested it yet.

- ❖ Integrate our proposed ChatWithRec real-time social recommendation system in one or more existing social network sites such as the Facebook platform to express its effectiveness and usefulness.
- ❖ Integrate an avatar (interactive online character) which might be a rich online social experience and make it easier. Therefore, we didn't consider this feature in our literature review but we will include it in future work.
- ❖ Conducted a case study experiment that contains a high number of users to enhance our system's efficacy and define the business benefits from the proposed SRS.
- ❖ Explore other common RSs problems like the data sparsity problem. Sparsity derives from the phenomenon that the number of rated items from users is less than the prediction model needs to build. That will be the focus of our future research problem.
- ❖ Explore the quality of service and security issues associated with recommender systems.
- ❖ Study more efficient and scalable TM techniques that can reduce the time needed to reach better results which helps improve the system performance.
- ❖ Build more reliable short textual databases to facilitate future data analysis.

References

Anette Hulth. (2003). “Improved automatic keyword extraction given more linguistic knowledge”. In Proceedings of the 2003 conference on Empirical methods in natural language processing (EMNLP '03). Association for Computational Linguistics, USA, 216–223. DOI:https://doi.org/10.3115/1119355.1119383

Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. (2002). “Getting to know you: learning new user preferences in recommender systems”. In Proceedings of the 7th international conference on Intelligent user interfaces (IUI '02). Association for Computing Machinery, New York, NY, USA, 127–134. DOI:https://doi.org/10.1145/502716.502737

Aditya. Anantharaman, Arpit. Jadiya, Chandana Tulasi Sai Siri; Bharath Nvs A, and Biju Mohan. (2019). "Performance Evaluation of Topic Modeling Algorithms for Text Classification". 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 704-708, DOI: 10.1109/ICOEI.2019.8862599

Agarwal, Vinti. And K. Bharadwaj. (2013). “A Collaborative Filtering Framework for Friends Recommendation in Social Networks Based On Interaction Intensity and Adaptive User Similarity”. Social Network Analysis and Mining 3(3) 359-379. DOI: 10.1007/s13278-012-0083-7

Alan R Hevner, March ST, Park J, Ram S. (2004). “Design science in information systems research”. Management Inform. Systems Quart. 28(1):75–105

Abbasi MA, Tang J, Liu H. (2014). “Trust-Aware Recommender Systems”. Machine learning book on computational trust. Chapman & Hall/CRC Press, Boca Raton

Ahmed I. Taloba, D. A. Eisa, and Safaa S. I. Ismail. (2018). “A Comparative Study on Using Principal Component Analysis With Different Text Classifiers”. International Journal of Computer Applications 180(31). DOI: 10.5120/ijca2018916800

Aiting Xu, Tianxing Qi and Xinzhi Dong. (2019). “Analysis of the Douban Online Review of the MCU: based on LDA topic model”. Journal of Physics: Conference Series, Volume 1437, 2nd International Symposium on Big Data and Applied Statistics (ISBDAS2019) 20–22 September 2019, Dalian, China. DOI:10.1088/1742-6596/1437/1/012102

References

Adomavicius. G and A. Tuzhilin, (2005). "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". In IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, June 2005, DOI: 10.1109/TKDE.2005.99

Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. (2014). “Facing the cold start problem in recommender systems”. Expert Systems with Applications 41, 4 (2014), 2065–2073. DOI: http://dx.doi.org/10.1016/j. eswa.2013.09.005

Behrooz Omidvar-Tehrani, Sruthi Viswanathan, Frederic Roulland, Jean-Michel Renders. (2020). “SAGE: Interactive State-Aware Point-Of-Interest Recommendation”. Workshop of State-based User Modelling, co-located with ACM International Conference on Web Search and Data Mining (WSDM) Conference, Houston Texas, USA, 4-7 February, 2020. SBN:978-1-4503-9999-9/18/06. https://doi.org/10.1145/1122445.1122456

Bayomi, Mostafa et al. (2019). “CoRE: A Cold-Start Resistant And Extensible Recommender System”. In Proceeding of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), Limassol, Cyprus

Benito-Picazo, F., Enciso, M., Rossi, C., Guevara, A. (2016). “Conversational recommendation to avoid the cold-start problem”. In: Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2016

Bridge, Derek and John Paul Kelly. (2006). “Ways of Computing Diverse Collaborative Recommendations”. In International Conference On Adaptive Hypermedia and Adaptive Web-Based Systems. Springer. 41-50. https://link.springer.com/chapter/10.1007/11768012_6

Brin, S. And L. Page (1998). “The Anatomy of a Large-Scale Hyper Textual Web Search Engine”. Computer Networks and ISDN Systems. VOL. 30(1). P:107-117. https://doi.org/10.1016/S0169-7552(98)00110-X

Burke, Robin. (2002). “Hybrid Recommender Systems: Survey and Experiments”. User Modeling and User-Adapted Interaction 12(4) 331-370. https://doi.org/10.1023/A:1021240730564

Burke Robin. (2007). “Hybrid Web Recommender Systems”. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. 377-408. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_12

Burke Robin, Kristian J. Hammond, and Benjamin C. Young. (1997). “The FindMe Approach to Assisted Browsing”. IEEE Expert: Intelligent Systems and Their Applications 12, 4 (July 1997), 32–40. DOI: https://doi.org/10.1109/64.608186

Candillier, L., Meyer, F., and Boullé, M. (2007). “Comparing State-of-the-Art Collaborative Filtering Systems”. In Machine Learning and Data Mining in Pattern Recognition, P. Perner, Editor, Springer Berlin Heidelberg, pp. 548-562. DOI:10.1007/978-3-540-73499-4_41

Clayton C. Peddy, D.A. (2003). “Building Solutions with Microsoft Commerce Server”. P. 656.

Chinchor, N. A. (1998). “Overview of Muc-7/MET-2”. Science Applications International Corporation 10260 Campus Pt. Dr. San Diego, CA 92121.

Chen A. (2005). “Context-Aware Collaborative Filtering System: Predicting the User’s Preference in the Ubiquitous Computing Environment”. In: Strang T., Linnhoff-Popien C. (eds) Location- and Context-Awareness. LoCA 2005. Lecture Notes in Computer Science, vol 3479. 978-3-540-25896-4. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11426646_23

Chen, J., Wang, C., Shi, Q., Feng, Y., & Chen, C. (2019). “Social recommendation based on users’ attention and preference”. Neurocomputing. 341. P: 1–9. DOI: https://doi.org/10.1016/J.NEUCOM.2019.02.045

Chughtai, M. W.;Selamat, A.; Ghani, I.;Jung, J. J. (2014). “E-Learning Recommender Systems Based on Goal-Based Hybrid Filtering”. IEEE International Journal of Distributed Sensor Networks. DOI: 10.1155/2014/912130

Daniel Ramage, Christopher D. Manning, and Susan Dumais. (2011). “Partially labeled topic models for interpretable text mining”. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11). Association for Computing Machinery, New York, NY, USA, P: 457–465. DOI: https://doi.org/10.1145/2020408.2020481

Dagan I., Glickman O., Magnini B. (2006). “The PASCAL Recognising Textual Entailment Challenge”. In: Quiñonero-Candela J., Dagan I., Magnini B., d’Alché-Buc F. (eds) Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment. MLCW 2005. Lecture Notes in Computer Science, vol 3944. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11736790_9

Dan, Wang. Xiang, Z. and Fesenmaier, D. R. (2016) ‘Smartphone Use in Everyday Life and Travel’, Journal of Travel Research, 55(1), pp. 52–63. DOI: 10.1177/0047287514535847

David Maio. Blei, Andrew Y. Ng. Michael I. Jordan. (2003). “Latent Dirichlet Allocation”. Journal of Machine Learning Research 3. 993-1022. https://doi.org/10.1002/cpe.4098

Deerwester, S. D. C., S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. (1990). “Indexing by Latent Semantic Analysis”. Journal of the American Society of Information Science. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

References

Derose, S. J. (1988). "Grammatical Category Disambiguation by Statistical Optimization". Computational Linguistics Association, Formerly the American Journal of Computational Linguistics, Volume 14, Number 1. 14(1) 31-39. https://www.aclweb.org/anthology/J88-1003

De Pessemier, T., Dooms, S., & Martens, L. (2014). "Context-aware recommendations through context and activity recognition in a mobile environment". Multimedia Tools and Applications, 72(3), 2925–2948. https ://doi.org/10.1007/s1104 2-013-1582-x

Dilip Singh Sisodia, Shivangi Bhandari, Nerella Keerthana Reddy and Abinash Pujahari. (2020). "A Comparative Performance Study of Machine Learning Algorithms for Sentiment Analysis of Movie Viewers Using Open Reviews". Performance Management of Integrated Systems and its Applications in Software Engineering, Asset Analytics, Springer Nature Singapore Pte Ltd. https://doi.org/10.1007/978-981-13-8253-6_10.

Esmaeili, L. , Minaei-Bidgoli, B. , Alinejad-Rokny, H. , & Nasiri, M. (2012). "Hybrid Recommender System for Joining Virtual Communities". Research Journal of Applied Sciences, Engineering and Technology, 4 (5), 500–509. ISSN: 2040-7467

Farzindar, A. And D. Inkpen (2015). "Natural Language Processing for Social Media": Morgan and Claypool Publishers. Link:
https://www.morganclaypoolpublisherscom/catalog_Orig/samples/9781681736136_sample.pdf

Fiky Hagar, Hussein Wedad, Gohary R.E. (2021). "Literature Review with Study and Analysis of the Quality Challenges of Recommendation Techniques and Their Application in Movie Ratings". In: Hassanien A.E., Darwish A. (eds) Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges. Studies in Big Data, vol 77. Springer, Cham. https://doi.org/10.1007/978-3-030-59338-4_12

Fedelucio, N. A. I. Pierpaolo Basile, Marco De Gemmis, Pasquale Lops, Giovanni Semeraro (2018). "A Domain-Independent Framework for Building Conversational Recommender Systems". Knowledge-aware and Conversational Recommender Systems (KaRS) Workshop 2018 (co-located with RecSys 2018), October 7, 2018, Vancouver, Canada.

Guy Shani and Asela Gunawardana. (2011). "Evaluating Recommendation Systems". In Recommender Systems Handbook, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, NY, USA. P. 257–297. http://link.springer.com/chapter/10.1007/978-0-387-85820-3_8

G. Jawaheer, P. Weller, and P. Kostkova. (2014). "Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback". ACM

Transactions on Interactive Intelligent Systems June 2014 Article 4(2):8:1–8. No.: https://doi.org/10.1145/2512208

Gambhir, M. And V. Gupta (2017). “Recent Automatic Text Summarization Techniques: A Survey”. Artificial Intelligence Review 47(1) 1-66. ntell Rev 47, 1–66. https://doi.org/10.1007/s10462-016-9475-9

Giammarino, D., Et Al. (2017). “Social Recommendation with Time and Sentiment Analysis”. Publication of The 25th Conference On User Modeling, Adaptation and Personalization - UMAP '17. 376-380. Adjunct Publication of the 25th Conference on User Modeling, Adaptation and PersonalizationJuly 2017 Pages 376–380. https://doi.org/10.1145/3099023.3099104

Gomez, C, Uribe and N. Hunt, (2015). “The Netflix Recommender System: Algorithms, Business Value, And Innovation” ACM Transactions on Management Information Systems, Vol. 6, No. 4, P. 1–19. Article No.: 13 https://doi.org/10.1145/2843948

Gregor, Shirley & Hevner, Alan. (2013). “Positioning and Presenting Design Science Research for Maximum Impact”. MIS Quarterly. 37. 337-356. 10.25300/MISQ/2013/37.2.01. https://doi.org/10.25300/MISQ/2013/37.2.01

Guo, B. Z. Yu Et Al. (2016). “Mobigroup: Enabling Lifecycle Support to Social Activity Organization and Suggestion with Mobile Crowd Sensing”. In IEEE Transactions On Human-Machine Systems. 46(3):1-13. DOI:10.1109/THMS.2015.2503290

Gunawardana A., Shani G. (2015). “Evaluating Recommender Systems”. In: Ricci F., Rokach L., Shapira B. (eds) Recommender Systems Handbook. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7637-6_8

Habibi. M and A. Popescu-Belis. (2015). "Keyword Extraction and Clustering for Document Recommendation in Conversations" in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 4, pp. 746-759, April 2015, DOI: 10.1109/TASLP.2015.2405482

Hofmann, T. (2001). “Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning”. 42, 177–196. https://doi.org/10.1023/A:1007617005950

Hutchins W.J. (2004). “The Georgetown-IBM Experiment Demonstrated in January 1954”. In: Frederking R.E., Taylor K.B. (eds) Machine Translation: From Real Users to Research. AMTA 2004. Lecture Notes in Computer Science, vol 3265. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30194-3_12

Hevner, A. R. and Chatterjee, S. (2010). “Design Research in Information Systems”. MIS Quarterly Vol. 28 No. 1, pp. 75-105. DOI:10.1007/978-1-4419-5653-8

References

Hofmann, T. (1999). “Probabilistic Latent Semantic Indexing”. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. Berkeley, California, USA, ACM, Pp. 50-57. https://doi.org/10.1145/312624.312649

H. Liang, Y. Xu, Y. Li, and R. Nayak. (2010). “Personalized Recommender System Based on Item Taxonomy and Folksonomy”. In Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM’10, P: 1641– 1644. ACM. https://doi.org/10.1145/1871437.1871693

Haoxian, Feng. (2017). “User Behavior Learning in Designing Restaurant Recommender Systems: An Approach to Leveraging Historical Data and Implicit Feedback”. Master Thesis. http://dx.doi.org/10.20381/ruor-21177

H. Ma, D. Zhou, C.Liu, M. R. Lyu and I. King. (2011). “Recommender Systems with Social Regularization”. In WSDM ’11.P. 287-296. Hong Kong, China. https://doi.org/10.1145/1935826.1935877

I. Akermi, M. Boughanem, and R. Faiz. (2016). “Just-in-time recommendation approach within a mobile context”. In 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016, 2016, pp. 636–639. DOI: 10.1109/WI.2016.0112

Isinkaye, F., Y. Folajimi and B. Ojokoh (2015). “Recommendation Systems: Principles, Methods and Evaluation”. V.16, Issue3, P: 261-273. https://doi.org/10.1016/j.eij.2015.06.005

Jiang, X, Y. Hu and H. Li. (2009). “A Ranking Approach to Key-Phrase Extraction”. The 32nd International ACM SIGIR Conference On Research and Development in Information Retrieval. https://doi.org/10.1145/1571941.1572113

Jolliffe, I. T. (1986). “Principal Component Analysis”. New York: NY. USA. http://cda.psych.uiuc.edu/statistical_learning_course/Jolliffe%20I.%20Principal%20Component%20Analysis%20(2ed.,%20Springer,%202002)(518s)_MVsa_.pdf

Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. (2013). “Addressing cold-start in app recommendation: latent user models constructed from twitter followers”. Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. Pages 283–292. Dublin, Ireland. https://doi.org/10.1145/2484028.2484035

Jaffali S., Jamoussi S., Khelifi N., Hamadou A.B. (2020) Survey on Social Networks Data Analysis. In: Rautaray S., Eichler G., Erfurth C., Fahrnberger G. (eds) Innovations for Community Services. I4CS 2020. Communications in Computer and Information Science, vol 1139. Springer, Cham. https://doi.org/10.1007/978-3-030-37484-6_6

References

J. Huang, X. Hu and C. Lu. (2011). "A comparison study on familiarity-based and similarity-based social networks," IEEE International Conference on Granular Computing, pp. 274-278, DOI: 10.1109/GRC.2011.6122607

J. Gomez, Boiy, E. & Moens, MF. (2012). "Highly discriminative statistical features for email classification". Knowledge and information systems. vol. 31. P: 23–53. https://doi.org/10.1007/s10115-011-0403-7

Jingu Kim, Yunlong He, and Haesun Park. (2014). "Algorithms for Nonnegative Matrix and Tensor Factorizations: a Unified View Based on Block Coordinate Descent Framework". Journal of Global Optimization 58, P. 285–319. DOI: https://doi.org/10.1007/s10898-013-0035-4

Jonsson, E. & Stolee, J. (2016). "An Evaluation of Topic Modelling Techniques for Twitter". https://www.semanticscholar.org/paper/An-Evaluation-of-Topic-Modelling-Techniques-for-J%C3%B3nsson/b3fcafa0d2a6e31078397e15a1ac3b700c1afea3

Johnson, W.B., Lindenstrauss, J. & Schechtman, G. (1986). "Extensions of lipschitz maps into Banach spaces". Israel J. Math. 54, 129–138. https://doi.org/10.1007/BF02764938

Jakob Nielsen. (1994). "Usability engineering. Elsevier". https://www.elsevier.com/books/usability-engineering/nielsen/978-0-08-052029-2

Kaplan, A. M. (2012). "If You Love Something, Let It Go Mobile: Mobile Marketing and Mobile Social Media". Business Horizons. V. 55(2). P: 129-139. https://doi.org/10.1016/j.bushor.2011.10.009

Kaplan, A. M. And M. Haenlein (2011). "Two Hearts in Three-Quarter Time: How to Waltz the Social Media/Viral Marketing Dance". Business Horizons. V. 54(3). P: 253-263. https://doi.org/10.1016/j.bushor.2011.01.006

Kautz, H., B. Selman and M. Shah (1997). "Referral Web: Combining Social Networks and Collaborative Filtering". ACM 40(3) 63-65. https://doi.org/10.1145/245108.245123

Ken Peffers, Tuure T, Marcus R, And Samir Chatterjee. (2007). "A Design Science Research Methodology for Information Systems Research". Inf. Syst. 24,45-77. DOI: https://doi.org/10.2753/MIS0742-1222240302

Keerthana, S. (2017). "Recommended Search of Documents from Conversation with Relevant Keywords Using Text Similarity". Journal of Network Communications and Emerging Technologies (JNCET). Volume 7, Issue 2. https://www.jncet.org/Manuscripts/Volume-7/Issue-2/Vol-7-issue-2-M-01.pdf

References

Kolini, F., Janczewski, L. (2017). “Clustering and Topic Modeling: A New Approach for Analysis of National Cyber Security Strategies”. In: Twenty First Pacific Asia Conference on Information Systems, Langkawi. http://aisel.aisnet.org/pacis2017

Kuroptev, Roman ; Lagerlöf, Anton. (2019).” Improving movie recommendations through social media matching”. Bachelor thesis. http://ls00012.mah.se/handle/2043/29240

Kandukuri, M., & V.V., H. (2021). “Topic Modelling Extraction of “Mann Ki Baat”. European Journal of Mathematics and Statistics, 2(1), 1-12. https://doi.org/10.24018/ejmath.2021.2.1.11

Kumar, K., Lu, Y.H.(2010). “Cloud Computing for Mobile Users: Can Offloading Computation Save Energy Computer”. 43(4), 51–56, DOI:10.1109/MC

Khan A., Ahmad A., Rahman A.U., Alkhalil A. (2020) A Mobile Cloud Framework for Context-Aware and Portable Recommender System for Smart Markets. In: Mehmood R., See S., Katib I., Chlamtac I. (eds) Smart Infrastructure and Applications. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-13705-2_12

Kamal, S., & Arefin, M. S. (2016). “Impact analysis of Facebook in family bonding. Social Network Analysis and Mining”. Social Network Analysis and Mining. V. 6(1), P: 1-14. DOI:10.1007/s13278-015-0314-9

Khater, S., D. Gracanin and H. G. Elmongui (2017). “Personalized Recommendation for Online Social Networks Information”: Personal Preferences and Location-Based Community Trends. IEEE Transactions On Computational Social Systems. vol. 4, no. 3, pp. 104-120, Sept. 2017, doi: 10.1109/TCSS.2017.2720632

Konstan, J. A. And J. Riedl (2012). “Recommender Systems: From Algorithms to User Experience”. User Modeling and User-Adapted Interaction. 22(1-2), P: 101-123. https://doi.org/10.1007/s11257-011-9112-x

Ken Peffers, Marcus R, Tuure T, And Reza Vaezi. (2012). “Design Science Research Evaluation”. In Proceedings of The 7th International Conference On Design Science Research in Information Systems: Advances in Theory and Practice (DESRIST'12), Springer, Berlin, Heidelberg, 398-410. DOI: Http://Dx.Doi.Org/10.1007/978-3-642-29863-9_29

Kytea. (2013). “Kyoto Text Analysis Toolkit”, Http://Www.Phontron.Com/Kytea/

Law, R., I. C. C. Chan and L. Wang (2018). “A Comprehensive Review of Mobile Technology Use in Hospitality and Tourism”. Journal of Hospitality Marketing & Management. https://doi.org/10.1080/19368623.2018.1423251

References

Li, C., Diao, Y., Ma, H., and Li, Y. (2008). "A statistical PCA method for face recognition," in Intelligent Information Technology Application, P: 376–380. DOI: 10.1109/IITA.2008.71

Lad, S. M. (2016). "Keyword Extraction from Conversation Text Document and Recommending Document Using Fuzzy Logic Based Weight Matrix Method". International Journal of Advanced Research in Computer Science. DOI: https://doi.org/10.26483/ijarcs.v7i4.2700

Lai, C.-H., Lee, S.-J., & Huang, H.-L. (2019). "A social recommendation method based on the integration of social relationship and product popularity". International Journal of Human-Computer Studies, 121, 42–57. https://doi.org/10.1016/J.IJHCS.2018.04.002

Lina Gong. W. Z. Tao Xu, Xuhong Li, Xia Wang, And Wenwen Pan. (2017). "The Application of Mixed Recommendation Algorithm with User Clustering in The Microblog Advertisements Promotion". In Conference Proceedings 1820, 080025. https://doi.org/10.1063/1.4977381

Liu, Z. Et Al. (2010). "Automatic Key Phrase Extraction Via Topic Decomposition". Proceedings of the 2010 Conference On Empirical Methods in Natural Language Processing, Association for Computational Linguistics. https://www.aclweb.org/anthology/D10-1036

Liu, L., Tang, L., Dong, W., S., Yao, and Zhou, W. (2016). An Overview of Topic Modeling and Its Current Applications in Bioinformatics. SpringerPlus. DOI: 10.1186/s40064-016-3252-8

Likhitha, S., Harish, B. S., and Keerthi Kumar, H. M. (2019). A detailed survey on topic modeling for document and short text data. Int. J. Comp. Appl. 178:39. DOI: 10.5120/ijca2019919265

Lucas Bernardi, Jaap Kamps, Julia Kiseleva, Melanie J.I. Mueller. (2015). "The Continuous Cold Start Problem in e-Commerce Recommender Systems". https://arxiv.org/pdf/1508.01177.pdf

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). "Facing The Cold Start Problem In Recommender Systems". Expert Systems with Applications, 41(4), 2065-2073. https://doi.org/10.1016/j.eswa.2013.09.005

Manning, C.D., Raghavan, P., and Schütze, H. (2009). "Introduction to Information Retrieval". Computational Linguistics. Vol. 35, MIT Press, 307-309. https://nlp.stanford.edu/IR-book/information-retrieval-book.html

M. R¨oder, A. Both, and A. Hinneburg, (2015). "Exploring the space of topic coherence measures". In Proceedings of the eighth ACM international conference on Web search and data mining WSDM '15. P. 399–408.https://doi.org/10.1145/2684822.2685324

M. Honnibal and I. Montani. (2017). "SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing". https://sentometrics-research.com/publication/72/

References

Mimno D, Wallach HM, Talley E, Leenders M, McCallum A. (2011). "Optimizing semantic coherence in topic models". In: Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, P: 262–272. http://dirichlet.net/pdf/mimno11optimizing.pdf

Mausam, Et Al. Jeju Island, Korea. (2012). "Open Language Learning for Information Extraction". Proceedings of The Joint Conference On Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics. 523-534. https://www.aclweb.org/anthology/D12-1048

March, S. T. And G. F. Smith (1995). "Design and Natural Science Research On Information Technology". Decision Support Systems. V. (15) P: 251-266. https://doi.org/10.1016/0167-9236(94)00041-2

Mazarura J, Waal A De, Kanfer F, Millard S. (2014). "Topic Modelling for Short Text". Proceedings of the PRASA, RobMech and AfLaT International Joint Symposium. Cape Town

Mccallum, A. K. (2002). "MALLET: A Machine Learning for Language Toolkit". http://mallet.cs.umass.edu/

Meng-Yen, K.-C. L. Tien-Hsiung. (2018). "A Keyword-Aware Recommender System Using Implicit Feedback On Hadoop". Journal of Parallel and Distributed Computing, Pages 63-73. https://doi.org/10.1016/j.jpdc.2017.12.008

Mihalcea, R., P. Tarau and E. Figa (2004). "PageRank On Semantic Networks, With Application to Word Sense Disambiguation". Proceedings of The 20th International Conference On Computational Linguistics, Association for Computational Linguistics. https://doi.org/10.3115/1220355.1220517

Maritza Bustos-López, Raquel Vásquez-Ramírez, Giner Alor-Hernández. (2015). "An Architecture for Developing Educational Recommender Systems". Res. Computer. Sci. 106: 17-26. article{DBLP:journals/rcs/Bustos-LopezVA15.

Nielsen, J. (2012). "How many test users in a usability study?". Retrieved from http://www.nngroup.com.

Nguyen, T. N. (2017). "Conversational Group Recommender Systems". Proceedings of The 25th Conference On User Modeling, Adaptation and Personalization - UMAP '17. Pages 331–334. https://doi.org/10.1145/3079628.3079704

Pascual, F. (2019). "Introduction to Topic Modelling". MonkeyLearn Blog. https://monkeylearn.com/blog/introduction-to-topic-Modelling/

References

Posner, Miriam. (2012). "Very Basic Strategies for Interpreting Results from the Topic Modeling Tool". Miriam Posner's Blog. https://miriamposner.com/blog/very-basic-strategies-for-interpreting-results-from-the-topic-modeling-tool/

Pan, Chenguang. And Wenxin. Li (2010). "Research Paper Recommendation with Topic Analysis". International Conference On Computer Design and Applications (ICCDA), IEEE. P. V4-264-V4-268, DOI: 10.1109/ICCDA.2010.5541170

Pandit, P. Umar Farooq Shahabad Et Al. (2016)." Keyword Extraction and Clustering for Document Recommendation in Conversations". International Journal of Advanced Research in Computer and Communication Engineering. P. 746-759. https://www.ijarcce.com/upload/2016/march-16/IJARCCE%20178x.pdf

Peng, Liu. Lemei Zhang and J. A. Gulla. (2019). "Real-Time Social Recommendation Based On Graph Embedding and Temporal Context". Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway. https://doi.org/10.1016/j.ijhcs.2018.02.008

P. Melville and V. Sindhwani, (2010). "Recommender Systems". In Encyclopedia of Machine Learning, S. Verlag, Ed., ed Berlin: Springer, P. 1-9

Phand, S.A and V. A. Chakkarwar. (2018). "Enhanced Sentiment Classification Using Geo Location Tweets" Second International Conference on Inventive Communication and Computational Technologies (ICICCT), IISC Banglore, India, P. 881-886, DOI: 10.1109/ICICCT.2018.8473048

Pinaki Prasad Guha Neogi, Amit Kumar Das, Saptarsi Goswami and Joy Mustafi. (2020). "Topic Modeling for Text Classification". Springer Nature Singapore Pte Ltd. Emerging Technology in Modelling and Graphics, Advances in Intelligent Systems and Computing 937, https://doi.org/10.1007/978-981-13-7403-6_36

P. J. Phillips et al. (2005). "Overview of the face recognition grand challenge". IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), P. 947-954. vol. 1, DOI: 10.1109/CVPR.2005.268

Preethi. R, Sheeja. S. (2021). "ERS: Latent Dirichlet Allocation Based E-Commerce Recommendation System Using Deep Neural Network". DOI:10.17762/pae.v58i2.2038

Quijano-Sanchez, L., C. Sauer, J. A. Recio-Garcia, and B. Diaz-Agudo. (2017). "Make It Personal: A Social Explanation System Applied to Group Recommendations". V. 76, Pages: 36-48. https://doi.org/10.1016/j.eswa.2017.01.045

References

Qian, Tieyun et al. (2019). “Solving Cold Start Problem in Recommendation with Attribute Graph Neural Networks”. ArXiv abs/1912.12398. DOI: 10.1109/TKDE.2020.3038234

Qiang, J., Zhen-Yu, Q., Yun, L., Yun-Hao, Y., & Xin-Dong, W. (2019). “Short Text Topic Modelling Techniques, Applications, and Performance: A Survey”. arXiv:1904.07695, Information Retrieval. DOI:10.1109/tkde.2020.2992485

Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, et al. (2020) “Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia”. N Engl J Med, 382 (13) (2020), P. 1199-1207. DOI: 10.1056/NEJMoa2001316

Qasim Ali Arain, ,Hina Memon, Imran Memon, and Muhammad Hammad Memon. (2017). “Intelligent travel information platform based on location base services to predict user travel behavior from user-generated GPS traces”. International Journal of Computers and Applications 39(3). DOI:10.1080/1206212X.2017.1309222

Rania Albalawi. (2019). “ChatWithRec recommendation system”. Https://github.com/Rania2016

Rania Albalawi, Tet Hin Yeap, And Morad Benyoucef. (2019). “Toward A Real-time Social Recommendation System”. MEDES’19, November 12-14, Limassol, Cyprus, (2019). https://doi.org/10.1145/3297662.3365789

Rania Albalawi, Tet Hin Yeap. (2019). “ChatWithRec: Toward A Real-time Conversational Recommender System”. ICCSMLBD, 174th-18th October. New York, USA. https://www.worldresearchlibrary.org/up_proc/pdf/3216-157319215067-71.pdf

Rania Albalawi, Tet Hin Yeap, And Morad Benyoucef. (2020). “Using Topic Modeling Methods for Short Text Data: A Comparative Analysis”. A Journal in the Frontiers in Artificial Intelligence-Machine Learning and Artificial Intelligence. DOI: 10.3389/frai.2020.00042. Manuscript Id: 539160

Rantanen, P. (2017). “Architecture for Interfacing Content Analysis Back Ends”. (Tampere University of Technology. Publication; Vol. 1460). Tampere University of Technology.

Rauch, G. (2012). “Smashing Node.js : JavaScript everywhere”. John Wiley & Sons Inc. Retrieved fromhttps://books.google.fi/books?hl=en&lr=&id=G1y_5kpmatUC&oi=fnd&pg=PT21&dq=nodejs&ots=uSo7T6SAZ1&sig=Decxu4wfVNJncNk8RC9FEWbogw&redir_esc=y#v=onepage&q=nodejs&f=false

Robertus Nugroho, Cecile Paris, Surya Nepal, Jian Yang, Weiliang Zhao. (2020). “A Survey of Recent Methods on Deriving Topics From Twitter”. Algorithm to Evaluation. Springer-Verlag

References

London Ltd. Springer. Knowledge and Information Systems. https://doi.org/10.1007/s10115-019-01429-z

Ramage, Daniel, Evan Rosen, Jason Chuang, Christopher D. Manning and Daniel A. McFarland (2009). "Topic Modeling for The Social Sciences". Stanford University Stanford, Ca 94305, Nips Workshop On Applications for Topic Models: Text and Beyond

Rehurek, R., Rehurek,R (2010). "Software Framework for Topic Modelling with Large Corpora". in Proceedings of the LREC Workshop on New Challenges for NLP Frameworks. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en

Reshma. Mr, S Neduncheliyan, And K Kalaivani. (2020). "A Survey On Multi- Media Hybrid and Collaborative Recommendation Based On Cloud". International Journal of Advanced Computer Technology (IJACT). VOLUME 4, NUMBER 1. https://www.ijact.org/ijactold/volume4issue1/IJ0410013.pdf

Resnick, P. And H. R. Varian (1997). "Recommender Systems". ACM. V. 40(3). P. 56-58.

Ricci, F. Rokach, L., Shapira, B., Kantor (2015). "Recommender Systems Handbook": ISBN 978-0-387-85820-3. Springer. https://link.springer.com/book/10.1007/978-0-387-85820-3

Revathy V.R., Pillai A.S. Iyer, B., Nalbalwar S., Pathak N. (2019). "A Proposed Architecture for Cold Start Recommender by Clustering Contextual Data and Social Network Data". In Computing, Communication and Signal Processing. Advances in Intelligent Systems and Computing, Vol 810. Springer, Singapore. https://doi.org/10.1007/978-981-13-1513-8_34

Rana C, Jain SK. (2012). "A Study of the Dynamic Features of Recommender Systems". Artif Intell Rev 43(1):141–153. DOI: https://doi.org/10.1007/s10462-012-9359-6

Rauch, G. (2012). "Smashing Node.js : JavaScript everywhere". John Wiley & Sons Inc. Retrieved:https://books.google.fi/books?hl=en&lr=&id=G1y_5kpmatUC&oi=fnd&pg=PT21&dq=nodejs&ots=uSo7T6SAZ1&sig=Decxu4wfVNJncNk8RC9FEWbogw&redir_esc=y#v=onepage&q=nodejs&f=false

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2001). "Generate models for cold start recommendations". Proceedings of the 2001 ACM SIGIR Workshop on Recommender Systems. ACM, New York. USA

Sara, A., Y. E. B. El Idrissi and R. Ajhoun (2016). "Time Aware Recommendation". The 6th International Conference On Information and Communication Technology for The Muslim World" (ICT4M), IEEE. DOI 10.1109 / ICT4M.2016.75

References

Santosh Kumar Ray, Amir Ahmad & Ch. Aswani Kumar. (2019). “Review and Implementation of Topic Modeling in Hindi, Applied Artificial Intelligence”, 33:11, 979-1007, DOI: 10.1080/08839514.2019.1661576. https://doi.org/10.1080/08839514.2019.1661576

S. Dudoit, J. Fridlyand, and T. P (2002). “Speed: Comparison of Discrimination Methods for The Classification of Tumor Using Gene Expression Data”. Vol. 97: P. 77-87. https://doi.org/10.1198/016214502753479248

Schafer, J. B., J. Konstan and J. Riedl (1999). “Recommender Systems in E-Commerce”. Proceedings of The 1st Conference On Electronic Commerce, ACM. https://doi.org/10.1145/336992.337035

Seo, Y.-D., Et Al. (2017). “Personalized Recommender System Based On Friendship Strength in Social Network Services”. Expert Systems with Applications. 69 135-148. https://doi.org/10.1016/j.eswa.2016.10.024

Sevli, Onur. E. U. Kucuksille. (2017). “Advertising Recommendation System Based On Dynamic Data Analysis On Turkish Speaking Twitter Users”, DOI: 10.17559/TV-20151020205558. https://hrcak.srce.hr/179876

Shu, K., Et Al. (2018). “Crossfire: Cross Media Joint Friend and Item Recommendations”. WSDM. Marina Del Rey, CA, USA. DOI: http://dx.doi.org/10.1145/3159652.3159692

Singh, B., Kushwah, S. Das and P. Johri. (2015). “Issue and Challenges of Online User Generated Reviews Across Social Media and E-Commerce Website”. ," International Conference on Computing, Communication and Automation(ICCCA). P. 818-822, DOI: 10.1109/CCAA.2015.7148486

Sohail, S. S., J. Siddiqui and R. Ali. (2017). “Classifications of Recommender Systems: A Review”. Journal of Engineering Science & Technology Review 10(4). DOI:10.25103/jestr.104.18

S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen. (2014). “Social Collaborative Filtering For Cold-Start Recommendations”. In Proceedings of the 8th ACM Conference on Recommender systems, P. 345-348. , Foster City, Silicon Valley, CA, USA. https://doi.org/10.1145/2645710.2645772

Sadia Zeb 1, Dr. Irfan Ali Bhacho, (2020). “Addressing Cold Start Item Problem in Recommender Systems” RMSE International Journal of Computer Science and Information Security(IJCSIS). Vol. 18, No.1. ISSN 1947-5500. https://sites.google.com/site/ijcsis/

S. Kumagai, T. Gokan, K. Tsubota, I. Isono, K. Hayakawa, et al. (2020). “Impact of the 2019 novel coronavirus on global economy: analysis using mobility data from mobile phones IDE

Policy". Brief No11 (April 28, 2020). https://www.ide.go.jp/library/Japanese/Publish/Download/PolicyBrief/IDE/pdf/11_en.pdf

Sheth, S., Arora, N., Murphy, C., & Kaiser, G. (2010). "weHelp: A Reference Architecture for Social Recommender Systems". IEEE/ACM International Conference on Automated Software Engineering workshops. Pt -160:461-472. PMID: 25285237; PMCID: PMC4181366

Stuart Edward Middleton. (2003). "Capturing knowledge of user preferences with recommender systems", Ph.D thesis, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science, University of Southampton

V. R. Revathy and S. Pillai Anitha. (2019). "Cold Start Problem in Social Recommender Systems: State-of-the-Art Review. Springer Nature Singapore Pte Ltd. Advances in Computer Communication and Computational Sciences, Advances in Intelligent Systems and Computing 759, https://doi.org/10.1007/978-981-13-0341-8_10

Volkovs, Yu, and Poutanen. (2017). "Dropout net: Addressing Cold Start In Recommender Systems". In 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. P: 4957–4966. https://www.cs.toronto.edu/~mvolkovs/nips2017_deepcf.pdf

Vulić. I, De Smet, W. Tang, J.& Moens, M. F. (2015). "Probabilistic Topic Modeling in Multilingual Settings: An Overview of Its Methodology and Applications". Information Processing & Management, 51(1), P: 111-147. https://doi.org/10.1016/j.ipm.2014.08.003

Vrishali Chakkarwar and Sharvari C. Tamane. (2020). "Quick Insight of Research Literature Using Topic Modeling". Springer Nature Singapore Pte Ltd. Smart Trends in Computing and Communications. Smart Innovation, Systems and Technologies 165, https://doi.org/10.1007/978-981-15-0077-0_20

Wang, D., S. Park and D. R. Fesenmaier (2012). "The Role of Smartphones in Mediating the Touristic Experience". Journal of Travel Research. V. 51(4). P: 371-387. https://doi.org/10.1177/0047287511426341

Wang, S., Et Al. (2017). "What Your Images Reveal: Exploiting Visual Contents for Point-Of-Interest Recommendation". Proceedings of The 26th International Conference On World Wide Web, International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3038912.3052638

Wang, P. Y., & Yang, H. C. (2012), "Using Collaborative Filtering to Support College Students' Use of Online Forum for English Learning", Computers & Education, V. 59(2), P. 628-637. https://doi.org/10.1016/j.compedu.2012.02.007

References

Wei, Y. Z., L. Moreau and N. R. Jennings (2003). “Recommender Systems: A Market-Based Design”. Proceedings of The Second International Joint Conference On Autonomous Agents and Multiagent Systems, ACM. DOI:10.1145/860575.860671

Witten, I. H., Et Al. (1999). KEA: “Practical Automatic Key Phrase Extraction”. Proceedings of The Fourth ACM Conference On Digital Libraries, ACM. DOI:10.1145/313238.313437

Wu, Chao. Y. Zhang, J. Jia and Wenwu. Zhu. (2017). “Mobile Contextual Recommender System for Online Social Media”. V. 16. P. 3403– 3416. In IEEE Transactions On Mobile Computing. DOI: 10.1109/TMC. .2694830

Wang, D., Ma, J., Lian, T., L., G. (2014). “Recommendation Based on Weighted Social Trusts and Item Relationships”. The 29th Annual ACM Symposium on Applied Computing (SAC). https://doi.org/10.1145/2554850.2554884

Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. Expert Systems with Applications, 69, 29–39. https ://doi.org/10.1016/j.eswa.2016.09.040

Wójcik, P.I. Kurdziel, M. (2019). “Training Neural Networks on High-Dimensional Data Using Random Projection”. Pattern Anal Applic 22,1221–1231(2019). https://doi.org/10.1007/s10044-018-0697-0

Wang. J, A. P. de Vries, and M. J. T. Reinders. (2006). “Unifying User-Based and Item-Based Collaborative Filtering Approaches By Similarity Fusion”. SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. P 12, 15. http://doi.acm.org/10.1145/1148170.1148257

Xu, Xiaoying, Kaushik, D. Anindye, D. Chunmian, Ge. (2018). “Identifying Functional Aspects from User Reviews for Functionality Based Mobile App Recommendation”. Journal of The Association for Information Science and Technology,V. 69(2). P: 242–255, Wiley Online Library (Wileyonlinelibrary.Com). https://doi.org/10.1002/asi.23932

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. (2013). “A Biterm Topic Model for Short Texts”. International World Wide Web Conference Committee (IW3C2). Rio de Janeiro, Brazil. ACM 978-1-4503-2035-1/13/05.

Yong Chen, Hui Zhang, Rui Liu, Zhiwen Ye, and Jianying Lin. (2019). “Experimental Explorations on Short Text Topic Mining Between LDA and NMF Based Schemes”. Volume 163, Pages 1-13. https://doi.org/10.1016/j.knosys.2018.08.011

References

Yadav, M., Y. Joshi and Z. Rahman. (2015). "Mobile Social Media: The New Hybrid Element of Digital Marketing Communications". Procedia-Social and Behavioral Sciences 189 335-343. https://doi.org/10.1016/j.sbspro.2015.03.229

Yuchen. Li, D. Zhang, Z. Lan and K. Tan. (2016). "Context-aware advertisement recommendation for high-speed social news feeding" IEEE 32nd International Conference on Data Engineering (ICDE), P. 505-516, DOI: 10.1109/ICDE.2016.7498266

Yuchen Li. (2016). "Real-Time Advertising On Social Networks". Thesis Submitted for The Degree of Doctor of Philosophy Nus Graduate School for Integrative Sciences and Engineering National University of Singapore.

Yera, R. And L. Martınez. (2017). "Fuzzy Tools in Recommender Systems: A Survey". International Journal of Computational Intelligence Systems. V. 10(1). P: 776-803. DOI: https://doi.org/10.2991/ijcis.2017.10.1.52

Y. Hu and M. Ogihara, (2011). "Nextone player: A music recommendation system based on user behavior". Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011

Zihuan, W. (2018). "A News-Topic Recommendation System Based on Keywords Extraction". In Kyusup Hahn Et Al. Eds., Springer US, 4339–4353. DOI:10.1007/s11042-017-5513-0

Zuckerberg, M. (2004). Facebook. Menlo Park, California, USA.

Appendix. 1

Method	Collaborative Filtering based RSs
Definition	-Based on users past behavior and interest like rating, past purchases, opinions, and decisions made by other users to predict future interest towards a certain item or opinion. It basically tries to find the similarity between users' profiles and items. -Focus on algorithms that match items or people preferences. -Has two approaches which are, user-based and item-based CF. -Mostly used in social networking sites.
Advantage	-Most used methods in the recommendation systems. -Doesn't rely on machine analyzable content in order to make accurate predictions (e.g System can make suitable recommendations on complex items without knowing any features of the item under discussion such as music). -Domain knowledge is not required.
Disadvantage	-Can't recommend items that are new or have not been previously rated as well recommend items to someone with unique tastes (tends to recommend popular items). -Scarcity -Cold start problem -Quality depends on large historical data.
Algorithm Used	Bayesian networks Clustering Graph modeling

	Pearson correlation K-NN approach Nearest neighbourhood classification
Example	Facebook, Uses collaborative filtering to recommend new friends and groups to the user
Method	**Content Based Filtering based RSs**
Definition	Uses keywords or other item's characteristics and features to recommend items and ideas with similar features. (Based on the item's content and user's profile rather than other user's opinions). -Use features from user's past liked items to infer new recommendations. -Focus on algorithms for learning user preferences and filtering a stream of new items. -Mostly used in E-commerce websites.
Advantage	-Ability to learn the users' preference of an item by using past contents or behavior on other items or features. -Reduce irrelevant content and provide users with more relevant products or services. -Domain knowledge is not required -The quality of recommendations improve over time (adaptive).
Disadvantage	-Only recommend items/content that the user has already experienced. -Over-specialization. -New user ramp-up problem.

	-Quality depends on large historical data. -Users who have very few ratings, would not be able to get accurate recommendations and it's hard to identify related interests of the same user because they predict items that are highly ranked in the user profile.
Algorithm Used	Cluster analysis Bayesian classifier techniques Neural networks Rocchio classification techniques Graph modeling Naive Bayes classifier Decision tree rule
Example	Radio stations like the Pandora Radio that uses initial song provided by the user as seed to recommend other songs with similar feature.
Method	**Hybrid filtering based RSs**
Definition	-Combine numerous Recommendation systems techniques to overcome the limitations of pure approaches and deliver better accuracy results. -Utilizes both the collaborative and content based filtering in order to predict the online behavior of the users in SM platforms in different ways such as, combining both of CF and CB features in order to optimize their strengths, making CF and CB predictions separately then combine them later, incorporating some of CF characteristics into CB method or vice versa, and mixing the features of CF and CB filtering into a general unified model.

Advantage	-Combine both CF and CB approaches advantages. -Good to predict the online behavior of the users in social media platforms. -Provide more accurate recommendations than the pure approaches.
Disadvantage	May include any of used filtering such as cold start problem in CF.
Algorithm Used	Techniques of both the collaborative and content based filtering applied with each other in different combinations. Linear combination of predicted ratings. Various voting schemes.
Example	Netflix, It gives recommendations by comparing the searching and watching patterns of similar users as well as by learning the searching and watching behavior of the user and giving recommendations of movies with similar features.
Method	**Knowledge filtering based RSs**
Definition	-Recommend item based on specific domain knowledge, by mapping certain item features and users' needs and preferences. -"Has knowledge about how particular items meet particular user needs, and employ this knowledge to recommend items based on inferences about each user's needs and preferences (Burke 2002) and (Burke 2007)."
Advantage	-Estimates how much user's requirements match the recommendations. -No ramp-up required. -Sensitive to changes of preference. -Effective with items that have a large number of features.

Disadvantage	-User must input utility function. -The drawback of all knowledge-based systems is the need for knowledge acquisition.
Algorithm Used	Machine learning Bayesian network AI techniques
Example	CATS, Collaborative Advisory Travel System that has been presented as a solution for the recommendation of holidays.
Method	**Context-Aware filtering based RSs**
Definition	A special kind of knowledge based system, when context is involved as knowledge, required for recommendation. Include location aware, temporal, trust aware and other contexts.
Advantage	CARS is necessary to understand the user’s delicate preferences and exploit the complications in their requirement explicitly.
Disadvantage	The user’s requirement is not static and may change time to time depending on many social and other factors affecting their purchasing trend.
Algorithm Used	User feedback AI techniques Machine learning Bayesian Networks
Example	LARS, Location-aware RS utilized location based ratings for recommendation places.

Method	Social Network filtering based RSs
Definition	Based on the preferences and rating provided by the user's friends and family members
Advantage	Following the rise of social-networks and uses the information about the social relations between users and the preferences of their friends and family.
Disadvantage	Large-scale Rich media Heterogeneity Ethical issues
Algorithm Used	Similarities measures. User profiling
Example	Facebook
Method	**Demographic filtering based RSs**
Definition	-Based on demographic filtering. -Tries to find the similarity between users' demographic information like, age, sex, occupation etc. -Categorize the user based on personal attributes and make recommendation based on demographic classes
Advantage	-Useful to define the category of users whose choices are similar for certain objects. -Uses the user's personal information to make predictions regarding their likes. -May not require a history of user ratings of the type needed by CF and CB techniques. -Recommendations quality improve over time.

Disadvantage	-New user/item ramp-up problem. -Must gather demographic information.
Algorithm Used	Correlation. Similarity measures.
Example	Book recommender system that use demographic data with knowledge based and content based techniques.