A hybrid recommender system for student academic advising supported by case-based reasoning and ontology

Charbel Obeid

N°d'ordre NNT : 2021LYSE1344

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

Ecole Doctorale ED512
Informatique et Mathématiques

Spécialité de doctorat :
Discipline : Informatique

Soutenue publiquement le 20/12/2021, par :
Charbel OBEID

A Hybrid Recommender System for Student Academic Advising Supported by Case-based Reasoning and Ontology

Devant le jury composé de :
Nom, prénom grade/qualité établissement/entreprise
CHENITI Lilia - Maître de Conférences - Université de Sousse (Tunisie) – Rapporteuse
MOUSSA Sherin – Professeure - Université Ain Shams - Le Caire (Egypte) – Rapporteuse
GHODOUS Parisa - Professeure des Universités - Université Lyon 1 - Examinatrice
KILANY Rima – Professeure - Université St Joseph – (Liban) – Examinatrice
MAKHOUL Abdallah - Professeur des Universités - Université de Franche-Comté (France) - Examinateur
LAHOUD Christine - Maître de Conférences - Université Française d'Egypte - Co-directrice de thèse
EL KHOURY Hicham - Maître de Conférences - Université Libanaise – (Liban) – Co-Directeur de thèse
CHAMPIN Pierre-Antoine - Maître de Conférences HDR - Université Lyon 1 (France) - Directeur de thèse

Université Libanaise

Ecole Doctorale
Sciences et Technologie

UNIVERSITÉ DE LYON

Lyon 1

N°d'ordre NNT : 2021LYSE1344

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

Ecole Doctorale ED512
Informatique et Mathématiques

Spécialité de doctorat :
Discipline : Informatique

Soutenue publiquement le 20/12/2021, par :
Charbel OBEID

A Hybrid Recommender System for Student Academic Advising Supported by Case-based Reasoning and Ontology

Devant le jury composé de :
Nom, prénom grade/qualité établissement/entreprise
CHENITI Lilia - Maître de Conférences - Université de Sousse (Tunisie) – Rapporteuse
MOUSSA Sherin – Professeure - Université Ain Shams - Le Caire (Egypte) – Rapporteuse
GHODOUS Parisa - Professeure des Universités - Université Lyon 1 - Examinatrice
KILANY Rima – Professeure - Université St Joseph – (Liban) – Examinatrice
MAKHOUL Abdallah - Professeur des Universités - Université de Franche-Comté (France) - Examinateur
LAHOUD Christine - Maître de Conférences - Université Française d'Egypte - Co-directrice de thèse
EL KHOURY Hicham - Maître de Conférences - Université Libanaise – (Liban) – Co-Directeur de thèse
CHAMPIN Pierre-Antoine - Maître de Conférences HDR - Université Lyon 1 (France) - Directeur de thèse

UNIVERSITE CLAUDE BERNARD - LYON 1

Administrateur provisoire de l'Université	M. Frédéric FLEURY
Président du Conseil Académique	M. Hamda BEN HADID
Vice-Président du Conseil d'Administration	M. Didier REVEL
Vice-Président du Conseil des Etudes et de la Vie Universitaire	M. Philippe CHEVALLIER
Vice-Président de la Commission de Recherche	M. Jean-François MORNEX
Directeur Général des Services	M. Pierre ROLLAND

COMPOSANTES SANTE

Département de Formation et Centre de Recherche en Biologie Humaine	Directrice : Mme Anne-Marie SCHOTT
Faculté d'Odontologie	Doyenne : Mme Dominique SEUX
Faculté de Médecine et Maïeutique Lyon Sud - Charles Mérieux	Doyenne : Mme Carole BURILLON
Faculté de Médecine Lyon-Est	Doyen : M. Gilles RODE
Institut des Sciences et Techniques de la Réadaptation (ISTR)	Directeur : M. Xavier PERROT
Institut des Sciences Pharmaceutiques et Biologiques (ISBP)	Directrice : Mme Christine VINCIGUERRA

COMPOSANTES & DEPARTEMENTS DE SCIENCES & TECHNOLOGIE

Département Génie Electrique et des Procédés (GEP)	Directrice : Mme Rosaria FERRIGNO
Département Informatique	Directeur : M. Behzad SHARIAT
Département Mécanique	Directeur M. Marc BUFFAT
Ecole Supérieure de Chimie, Physique, Electronique (CPE Lyon)	Directeur : Gérard PIGNAULT
Institut de Science Financière et d'Assurances (ISFA)	Directeur : M. Nicolas LEBOISNE
Institut National du Professorat et de l'Education	Administrateur Provisoire : M. Pierre CHAREYRON
Institut Universitaire de Technologie de Lyon 1	Directeur : M. Christophe VITON
Observatoire de Lyon	Directrice : Mme Isabelle DANIEL
Polytechnique Lyon	Directeur : Emmanuel PERRIN
UFR Biosciences	Administratrice provisoire : Mme Kathrin GIESELER
UFR des Sciences et Techniques des Activités Physiques et Sportives (STAPS)	Directeur : M. Yannick VANPOULLE
UFR Faculté des Sciences	Directeur : M. Bruno ANDRIOLETTI

Acknowledgments

I would like to express my gratitude and appreciation to my supervisors, HDR Dr. CHAMPIN Pierre-Antoine, MCF LAHOUD Christine, and MCF AL KHOURY Hicham for their constant support, encouragement, guidance, and advice throughout this study. It was a great honor to work with them for this work.

I would also like to thank Claude Bernard University Lyon 1 and the Lebanese University for offering me the opportunity to pursue doctoral studies. This opportunity has given me a meaningful experience. I have always felt grateful for this opportunity.

I am greatly indebted to my family especially my mother OBEID Jamal, and my wife OBEID Rita for their support, patience, and help. I would not have been able to complete this thesis without their support and help. Also, my fondest love and thanks to my deceased father OBEID Youssef. His spirit will be with me forever.

Finally, I would like to thank the members of my jury HDR CHENITI Lilia, Prof. MOUSSA Sherin, Prof. GHODOUS Parisa, HDR KILANY Rima, and Prof. MAKHOUL Abdallah for reviewing this thesis.

OBEID Charbel

Abstract

Academic advising at most high schools is limited in its ability to help students in identifying appropriate academic pathways. For example, choosing a university and a university major is a challenging task rife with anxiety that gets students confused. Therefore, students at high school need assistance to match their interests with the available universities and university majors. This thesis presents a novel hybrid recommender system (RS) approach for students' academic advising supported by *Case-based reasoning (CBR)* and *ontology*. This *Knowledge-based (KB)* hybrid *RS* enables high school students to submit their demographic data, courses' ratings, preferences, interests, and other knowledge in order to get personalized recommendations. These recommendations consist of three essential suggestions namely the university/college, university major, and career choice that match the student interests and requirements. The five major contributions of our research are: (1) Proposing a novel hybridization approach that incorporates four-core technologies namely the *KB* recommender system*, Collaborative Filtering (CF), CBR,* and *ontology;* (2) Reducing the limitations and problems of the basic recommender systems' techniques; (3) Designing the ontology that describes the high school student's profile and interests, high school, university/college, and career domains; (4) Generating personalized recommendations based on the case-base and ontology; (5) Treating through the proposed hybrid RS high dimensional datasets that contain heterogeneous data types. A prototype of this hybrid *RS* named (COHRS) is developed and presented. The experimental assessments and comparisons with the existing traditional recommender systems' approaches, strongly demonstrate the high performance and accuracy of our hybrid *RS* that is supported by *CBR* and ontology.

Keywords: Hybrid Recommender Systems, Similarity Metrics, Case-based Reasoning, Ontology, Domain Knowledge Modelling, Clustering Techniques, university path, high school students, education.

Résumé

L'orientation académique dans la plupart des écoles secondaires est limitée dans sa capacité à aider les étudiants à identifier les parcours académiques appropriés. Par exemple, le choix d'une université et d'un domaine d'études est une tâche difficile, pleine d'anxiété, qui rend les étudiants confus. Par conséquent, les étudiants du secondaire ont besoin d'aide pour faire correspondre leurs intérêts avec les universités et les domaines d'études disponibles. Cette thèse présente une nouvelle approche de système de recommandation hybride (RS) pour l'orientation académique des étudiants, soutenue par le raisonnement basé sur les cas (CBR) et l'ontologie. Ce système hybride basé sur les connaissances (KB) permet aux lycéens de soumettre leurs données démographiques, leurs évaluations de cours, leurs préférences, leurs intérêts et d'autres connaissances afin d'obtenir des recommandations personnalisées et précises. Ces recommandations consistent en trois suggestions essentielles, à savoir l'université, le domaine d'études universitaires et le choix de carrière qui correspondent aux intérêts et aux exigences des lycéens. Les cinq contributions majeures de notre recherche sont: (1) Proposer une nouvelle architecture d'hybridation qui incorpore quatre technologies de base, à savoir le système de recommandation KB, le Filtrage Collaboratif (CF), le CBR et l'ontologie; (2) Surmonter les limites et les problèmes des techniques de base des systèmes de recommandation; (3) Concevoir l'ontologie qui modélise le profil et les intérêts des lycéens, l'enseignement supérieur et les domaines de carrière; (4) Générer des recommandations personnalisées et précises basées sur la base de cas et l'ontologie; (5) Traiter via notre système hybride des ensembles de données de grande dimension ayant des types de données hétérogènes. Un prototype de ce système de recommandation hybride nommé COHRS est développé et présenté. Les évaluations expérimentales et les comparaisons avec les approches des systèmes de recommandation traditionnels existants, démontrent fortement la performance et la précision de notre système hybride basé sur le CBR et l'ontologie.

Mots-clés: Systèmes de recommandation hybrides, métriques de similarité, raisonnement basé sur des cas, ontologie, modélisation de la connaissance du domaine, techniques de clustering, parcours universitaire, lycéens, éducation.

Table of contents

List of figures

List of tables

List of abbreviations

AI – Artificial Intelligence

CBR – Case-based Reasoning

COHRS – CBR and Ontology based Hybrid Recommender System

CF – Collaborative Filtering

CB – Content-based

DF – Demographic Filtering

DL – Descriptive Logic

FCA – Formal Concept Analysis

KB – Knowledge-based

ML – Machine Learning

NLP – Natural Language Processing

NN – Nearest Neighbor

OB – Ontology-based

OWL – Web Ontology Language

RS – Recommender System

RDF – Resource Description Framework

RDFs – RDF Schema

RMSE – Root Mean Square Error

RIF – Rule Interchange Format

SW – Semantic Web

SKOS – Simple Knowledge Organization System

SPARQL – SPARQL Protocol and RDF Query

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

XML – Extensible Markup Language

PART I

INTRODUCTION

CHAPTER 1: Introduction

This chapter presents the thesis background, research questions and objectives, research importance and contributions, and structure of the thesis.

1.1 Background

The transition from high school to university is a major shift in the life of a student and can be one of the most challenging experiences. It is challenging for not only academic purposes but also for financial, emotional, and social problems. This transition can be problematic in many forms like student's anxiety, adjustment processes, and continuity with respect to the curriculum (David H., 1996). Continuity with respect to subject studies can be problematic because students are not aware of the difference of the subject studied at school and university, in terms of workloads and learning and assessments style. Thus, there will be major changes that will have to be made by high school students such as adjustments to new learning styles, assessments styles, and writing styles. In addition, the university has a more difficult and comprehensive environment than schools. Therefore, when students join their first month in the university, they will discover that the expectations they had prior are actually the other way around. If the mentioned problems continued, this might lead to dropping out of the student from university major.

Many students have failed to adapt the academic transition when they join university because they only have a basic knowledge about university life. Many high school students lack broad knowledge of the available university disciplines' suitability to their own interests and preferences. For example, many students register in prestigious academic major such as engineering and medical faculties, find later that those programs don't match their expectations. When first year students find that their selected discipline is different to what they planned, they become dissatisfied and discouraged. This is all because of the big gap in awareness and guidance at the level of public and private schools. To avoid separation from the university, students seek to come into university with a strong high school curriculum background. No one helped students to discover university majors, subjects, career descriptions and salaries. Students arrive ill-prepared for studying at university, where teaching usually takes place in large room sizes, where learners are taught by instructors who are involved in a diversity of other roles (Hassel and Ridout, 2018). (Tett et al., 2017), stated that students at the first year university revealed the loss of a sense of belonging and at the end of the first semester, most students felt unclear about what was expected of them.

On the other hand, (V.N., 2007) revealed that 20% to 50% of students in the United States start their university journey with an undecided major and 50% to 75% of learners in higher education have changed their major at least once before graduation. This suggests that students' career choices are unclear upon university admission and enrollment.

(Tett et al., 2017), (Siri et al., 2016) outlined the transition from high school to university by socio-cultural perspectives affected by personal factors and the learning environment, comprising students' previous experiences. A complex integration between individual characteristics in terms of parental, social, educational background, and educational environment offered by universities affect transitions. The transition includes adolescents in transfer to adulthood, stressed to start a

new learning system that necessitates flexibility, self-organization, and self-regulation (Siri et al., 2016).

For instance, in France many high school students are involuntarily directed to pursue a university major assigned by a software. This is also stripping students from their choices, which could lead them to drop out of the university or major (Goux, D. et al., 2017).

Besides, in most Arab University, admissions are based on high-school results. Thus, students are distributed among university academic departments according to their exams' results. This distribution system causes students to study subjects that they are not interested in. This admission system does not take into account students' interests, habits, or performance in high school.

Also, a study of the Lebanese baccalaureate Curriculum shows that having a rigorous high school program and no clear career guidance, make students ill-prepared at university (Khoury, 2020). In the overwhelming school program, Ghiey A., the vice-president at the Lebanese American University, in interview in L"Orient le Jour (Khoury, 2020), stated that high school students do not have time to think of what to study at the university or recognize what is happening outside their schools.

Moreover, in Lebanon career counseling does not exist in schools and universities, except for a few private institutions. Unfortunately, the orientation programs in most schools are not well designed to cater to students' varied needs. (Vlaardingerbroek et al., 2017) stated that guidance is just a theory, which has nothing to do in practice. A small number of private schools have established career guidance offices, and some schools referred to private organizations such as Waznat and Labora (career guidance organizations). Some schools provide career guidance by means of an annual career fair. In an interview in L"Orient le Jour Ghiey A., (Khoury, 2020) stated that the presentations given by some universities' representatives in schools were not useful. Whereas, some schools do not allow university representatives.

In addition to that, the complexity of life and the instability of the job market strongly affects Lebanese youth when choosing a field of study. Faced with these problems, Lebanese students feel lost when choosing their university majors. Furthermore, the financial struggle is worse today in Lebanon due to the economic, political, and Covid-19 issues. It is not to be ignored that culture and traditions in Lebanon play role in career guidance. Some students are affected by their parents in their choices and directed to reach prestigious jobs. In some cases, tradition requires the eldest in the family to select medicine or engineering majors regardless of his/her interest**.**

Thus, the *World Wide Web (WWW)* could be a significant tool for helping students addressing the mentioned problems by providing them useful information related to universities/colleges, university majors, careers market, etc. However, the explosive evolution of data on the Web network with the growth of innovative electronic machines has made the *WWW* information increasingly significant in most internet users' life. However, internet users are forced sometimes

to take inappropriate decisions when searching the Web due to an incapability to deal with unstructured and massive volumes of data.

Therefore, due to the weak academic advising systems in most schools and the inability to cope with the massive volumes of information on the Web, students need assistance to explore universities/colleges and university majors that match their interests and preferred careers. Thus, this research focuses on developing a novel hybrid RS that enables students to explore top N recommendations based on their fields of interest. This hybrid RS main role is to assist learners in making the right decision when selecting their university/college, university major and career domain.

A RS is a software that provides personalized and appropriate recommendations to users (Kantor, P. B et al., 2011). The main purpose of a RS is to predict items that are probably preferred by a user based on his/her preferences and tastes. This type of system address information overload and help users take decisions suitable to their needs and interests. When using a RS, users will have the option to select which product to buy, which movie to watch, or which article to read. Such software is commonly used when the volume of online data outperforms any user's capability to analyze it. Data is integrated implicitly or explicitly into recommender systems. Users' clickstreams and Web navigation are considered as implicit data for a RS while users' ratings, evaluations, and feedback are considered as explicit data. Recommender systems have been used in many fields such as e-commerce, movie, book, music, tourism, hotel, e-learning, and medicine. Therefore, it can be a potential tool in guiding high school students when choosing appropriate universities/colleges and university majors that align with their aspiring careers.

However, recommender systems have many limitations in recommending precise choices of educational materials. These limitations occur due to the variety in the studying style and education level of the learners (Jhon K. Tarus et al., 2017). Traditional RS such as CF bases their recommendations on users' ratings or evaluations. However, studies reveal that these systems experience many issues in their recommendation processes such as *cold-start*, *sparsity, etc.* problems (Bobadilla, J. et al., 2012). The *cold-start* problems occur when the dataset does not include sufficient ratings and preferences. Therefore, reliable recommendations will become hard to provide. Whereas, the sparsity problem occurs when the items and users matrix table is widely sparse. In this case, the precision of the recommendations will decrease since past users could not rate all the available products in the system. Furthermore, traditional RS does not take into consideration the supplementary knowledge about the products and the users when generating the required recommendations (G. Adomavicius et al., 2011).

In order to enhance traditional recommender systems filtering techniques and reduce their limitations, the ontology and CBR system are incorporated within the recommender engine. Ontology is a formal description of knowledge as a collection of classes within a specific domain

and the relations that hold between them (Staab and Studer, 2004). The conceptual model of the ontology permits the reasoning at all concept levels. Recently, ontologies have attracted extensive attention in designing domain knowledge of courses, e-learning, news, software engineering, etc. Ontology is integrated into our hybrid recommender system to model the knowledge of the domain, users, and items. Ontology is used to describe the knowledge of the education domain, career domain, students' profiles, and university graduates' prior cases. In addition, to find similarities between concepts, ontology is used to compute the semantic relationship.

Besides, CBR (Perner, 2019) is an Artificial Intelligence (AI) technique applicable to problem-solving and learning where earlier cases are available. CBR is the process of addressing a new problem based on the solutions of similar prior problems. Solutions for the current problem are retrieved from a library of prior cases called case-base. Our hybrid system integrates the CBR approach to solve high school current cases. The CBR system solves new cases and generates solutions based on retrieving similarities from the case-base. Every successful and unsuccessful solution will be saved in the case-base and used for future case solving.

Generating effective recommendations to high school students with fewer actions from those users is a hard work research subject. The work presented in this study has been devoted to designing and developing an efficient RS. Several challenges, as defined below, have been occurred in the development of the novel hybrid RS:

- Design the domain ontology.
- Compute through the recommendation engine heterogeneous data types such as nominal, ordinal, numerical, etc.
- Treat high-dimensional datasets that contain more than 50 attributes.
- Select an optimal filtering model and a similarity metric for the RS that is based on ratings.
- Incorporate the *CF* and *KB* recommender system techniques in a uniform hybrid system.
- Integrate the ontology and *CBR* system into the *KB* system.
- Generate personalized recommendations based on high school students' ratings and demographic data and domain knowledge.

In the next sections, the research questions and objectives, research importance and contributions, and structure of the thesis are presented.

1.2 Research questions and objectives

In order to address the problems presented in the above section, this research answers the following questions:

- What are the appropriate recommendation techniques to process high school students' and university graduates' interests, education, demographics, and career knowledge to generate personalized recommendations?
- What are the suitable hybridization techniques to incorporate the KB and CF engines and interconnect them in a uniform hybrid RS?
- What main recommendation improvement comes from integrating the ontology into the KB engine, and how can this integration enhance object matching?
- What main recommendation enhancement comes from integrating the CBR technique into the KB engine, and how to interconnect it with the ontology to increase the accuracy of recommendations?
- What are the appropriate techniques to integrate heterogeneous data types into the hybrid recommendation process, and how the system can treat these types of data?
- What are the applicable solutions to overcome the limitations of traditional RS techniques and what are the fitting treatments to treat high dimensional datasets via the hybrid RS engine?

By answering the research questions, this study has attained the following five objectives:

The first objective: Generate to high school students personalized recommendations related to higher education path.

The recommendation process starts by processing the high school courses' ratings that are obtained from the student input. Then, a second input phase accepts the student query that describes his/her interests and demographic data. The ratings and queries entered by the student are used in the recommendation process in order to retrieve the similarity between the current student and prior graduates' cases based on a CBR system. Thus, the high school student gets appropriate and personalized recommendations based on his/her queries and ratings. These recommendations cover three main areas namely the university, university major, and career domain.

The second objective: Propose a novel hybrid RS approach that incorporates two core systems namely the KB and CF.

A novel knowledge-based hybrid RS approach has been proposed. With this hybrid system, we have benefited from the essential features of four technologies namely the KB, CF, CBR, and Ontology. This hybridization technique enables the use of CF and KB together in a uniform system to recommend personalized recommendations. With the support of ontology and CBR, the recommender engine of this hybrid system performed effectively in retrieving similarities between the domain objects.

The third objective: Overcome the limitations of basic recommender systems' techniques.

In order to overcome the limitation of the traditional RS techniques, a hybridization strategy has been developed. This hybridization strategy combines the KB and CF techniques in a single system. The CF main role is to compute ratings and generate recommendations. The KB system uses the recommendations of the CF, domain ontology, and the graduates' case-base to generate personalized recommendations to the high school student. This collaboration strategy between the KB and CF recommender systems is based on the "Feature augmentation" technique (Bruke, R., 2002).

The fourth objective: Integrate the *ontology* and *CBR* into the recommendation process of the hybrid RS.

The developed hybrid RS generates recommendations to active users based on the ontology domain and CBR system. The integration of the ontology in the hybrid system engine showed a significant role in the recommendation process. The ontology is used to model the knowledge of the domain, users, and items. The ontology is integrated into the proposed system to present the knowledge of education domain, career domain, students' profiles, and university graduates' prior cases. To find the similarity between concepts, the ontology is needed to compute the semantic relationship. In addition, the conceptual model of the ontology permits the reasoning at all concept levels.

Moreover, this hybrid system integrates the CBR approach to solve high school current cases. CBR is a technique used for mixing problems resolving and case learning. The CBR system solves new cases and generates solutions based on retrieving similarities form the case-base. Every successful and unsuccessful solution will be saved in the case-base and used for future case solving.

The fifth objective: Address the challenges of treating high-dimensional datasets that contain heterogeneous data types.

To fuel our RS, we have gathered data from the university graduates. The collected data encompasses more than 50 attributes that describes graduates' preferences and their academic trajectory. To tackle the high dimensionality of the 50 attributes, we have proposed a segmentation process to divide the dataset into 3 main categories, namely the ratings, domain knowledge, and demographic data. This technique helped us to integrate the available categories into the appropriate recommender system engines. For example, the ratings have been integrated into the CF system whereas the domain knowledge and the demographic data have been integrated into the KB system.

1.3 Research importance and contributions

By addressing the five declared objectives, this research contributes to recommender systems and the ontology in the following aspects:

(1) It proposes a new hybridization architecture that encompasses four-core technologies namely the KB, CF, CBR, and ontology. The CF interconnects with the KB system in order to generate personalized and appropriate recommendations to high school student. This hybrid system is supported by the ontology and CBR system that enhance the recommendations through the ontology similarity and the CBR case matching.

(2) It proposes a novel KB hybrid RS consisting of two main phases: the first phase encompasses the data collection, data preprocessing, and domain ontology development. whereas, the second phase encompasses four process steps namely the CF recommendation generation, CF recommendation integration into the KB system, KB recommendation generation based on ontology and CBR system, new case saving in the case-base of the CBR system.

(3) It proposes a domain ontology that describes higher education, career fields, and students' knowledge. This ontology describes the school and university/college institutions, fields of study, jobs' domains, and details about students' models.

(4) It reduces the limitations of the traditional RS techniques. By combining the CF and the KB recommender techniques into a single hybrid system, the CF limitation has been addressed. Here, the students' courses' ratings have been integrated into the CF recommendation process, whereas the users' knowledge and demographic data have been integrated into the KB recommendation process.

(5) It treats high dimensional datasets and integrates heterogeneous data types into a uniform hybrid RS. This system deals with nominal, ordinal, ratings, and numerical data types in order to generate precise recommendations.

(6) It demonstrates the weaknesses and strengths of some clustering techniques such as the *FCA, K-modes, and Hierarchical.* The *FCA* analysis revealed its weaknesses in clustering high dimensional datasets whereas the *K-modes* performed better and faster than the *Hierarchical* technique in clustering the users' trajectories. However, the *K-modes* technique has been suspended from our project and postponed for future research, and the CBR and ontology have been applied to the proposed hybrid RS.

1.4 Structure of the thesis

This thesis is structured into six chapters, which include a detailed review of ontology, CBR, and recommendation techniques in the context of recommender systems, followed by the proposed novel hybrid RS approach. Specifically, chapters 2 provides the background to ontology, CBR, recommender systems techniques, and related works. Chapter 3 present several data mining techniques for data pre-processing. Chapter 4 presents the first proposed approach based on

clustering techniques. Chapter 5 proposes the second approach which is a KB hybrid recommender system based on CBR and ontology. Chapter 6 implements and evaluates several recommendation techniques. Finally, the conclusion section summarizes the thesis and discusses our perspectives and future work. The following is a descriptive list of the seven chapters:

Chapter 1: presents the topic of this research that is the recommender systems using a novel hybridization approach supported by ontology and CBR systems. The first part of this chapter starts with a description of the background to the topic and then lists the research's main objectives and contributions.

Chapter 2: presents the *Semantic Web* standards, ontology structure, and main roles of ontology. Moreover, this chapter presents the CBR features and the generic *CBR* cycle. Furthermore, this chapter includes a significant section that explains in details the recommender systems concept and types. In this section, firstly, the data required for recommender systems are illustrated. Secondly, this chapter reviews the basic types of RS techniques such as KB, CF, Content-based (CB), Demographic Filtering (DF), and hybrid and presents their limitations. Thirdly, the seven hybridization strategies are discussed. Fourthly, the similarity metrics, neighborhood-based CF algorithms, and evaluation metrics are discussed. Fifthly, a literature review of the related areas is presented. Finally, a summary table of the advantages and disadvantages of the basic recommender systems techniques is showed.

Chapter 3: describes the preparation and dissemination of our online survey. To achieve our study, we worked on disseminating an online survey that includes more than 50 attributes. This survey's purpose is to reach university graduates and collect knowledge about their education trajectories and current career occupation. Additionally, this chapter presents our work in the data-preprocessing phase using techniques such as the *WordNet* and *Levenshtein distance*. The two techniques were implemented and tested in R language in order to clean and refine the collected dataset.

Chapter 4: presents details of the implementation and analysis of the *Formal Concepts Analysis (FCA)* clustering technique. Firstly, the *Concept Lattice* and *Conceptual Clustering Process* are discussed. Secondly, the FCA experiments and evaluation are demonstrated. Thirdly, the *Hierarchical* and *K-modes* for data clustering are reviewed. These latter techniques were implemented in order to cluster the mixed data of the collected dataset.

Chapter 5: this chapter proposes a novel hybrid RS approach supported by ontology and CBR system. The hybrid system architecture comprising the KB and CF techniques. Additionally, this chapter lists the 4 main phases of the proposed hybrid system, which begins with the data

acquisition, data preprocessing, ontology development, and ends with the development of the recommender systems.

Chapter 6*:* implements and evaluates four *RS* techniques specifically the (*CF, DF, KB*, and *Hybrid RS*). The experiments and evaluations of the mentioned RS techniques have covered five recommendation strategies namely the (Stand-alone user-based/item-based *CF* technique, Stand-alone *DF* technique, Stand-alone *KB* technique supported by *CBR*, Stand-alone *KB* Technique supported by CBR and ontology, and *Hybrid KB* with *user-based CF* techniques supported by *CBR* and *ontology*. Finally, a comparative analysis of the tested recommender systems is discussed.

Finally, the conclusion recapitulates the thesis. Our perspectives, future research, and improvements are then discussed.

PART II

RELATED WORKS

CHAPTER 2: Literature review and related works

This chapter introduces the *Semantic Web (SW)* technologies and ontology, an overview on the CBR systems, the concept lattice and FCA, details information about the recommender systems, and related works.

2.1 Introduction

This chapter presents in detail the concepts and technologies that are used in our research to increase the efficiency of the proposed hybrid RS. These technologies are the Semantic Web, Ontology, CBR, FCA, recommender systems, etc. By implementing and testing these technologies, we achieved our aim to provide high school students appropriate recommendations toward higher education paths. In addition, this chapter presents the literature review and related works of the mentioned technologies.

The *SW* provides a solution for the machine to treat online data. The *SW* (Hitzler, 2021) is an extension of the current *World Wide Web* (WWW) that offers programmable applications with machine-interpretable metadata of the online data. The *SW* adds extra data descriptors to available content on the Web. This enables machines to make meaningful interpretations the same way people analyze data to make useful decisions.

The *ontology* is a formal description of knowledge as a collection of classes within a specific domain and the relations that hold between them (Staab and Studer, 2004). Recently, ontologies have attracted extensive attention in designing domain knowledge of courses, e-learning, news, software engineering, etc.

The *Case-based reasoning* (CBR) (Perner, 2019) is an *Artificial Intelligence* (AI) technique applicable to problem-solving and learning where earlier cases are available. CBR is the process of addressing a new problem based on the solutions of similar prior problems. Solutions for the current problem are retrieved from a library of prior cases called case-base.

The *Formal Concept Analysis* (FCA) (Varga et al., 2016) is a mathematical concept. FCA is a significant technique within the information retrieval domain and concept formalization. This technique is implemented in several fields specifically in machine learning, data mining, data preprocessing, and ontology construction.

The recommender systems (Aggarwal, 2016a) are the most significant machine learning tools, which predict users' behaviors and recommend personalized items such as *courses, articles, books, movies, playlists, etc.* The origin of RS is based on information retrieval. In addition, this type of system involves different intelligent techniques such as text analysis, text mining, semantic technologies, artificial intelligence, machine learning, etc. The next section, introduces the SW concept and standards.

2.2 Literature review

2.2.1 Semantic web

Petabytes of data are published online and available for all the internet users. However, the data are not understandable by the machine. Besides, the HTML pages present the online data in many different formats that are difficult for machines to process. Here comes the major role of the *SW*, which provides a solution for the machine to treat online data. The *SW* (Hitzler, 2021) is an extension of the current WWW that offers programmable applications with machine-interpretable metadata of the online data. The *SW* adds extra data descriptors to available content on the Web. This enables machines to make meaningful interpretations the same way people analyze data to make useful decisions. The motivations for applying the *SW* technologies to the Web comes under the umbrella of three factors: Individual Assistants, Automated Information Retrieval (AIR), and the Internet of Things (IoT). Thus, the *SW* represents the next major evolution in linking knowledge through semantic relationships. These relationships enable Web data to be linked and understood by machines in order to compute sophisticated tasks on users' behalf. So, the Web is directed toward a new phase of development.

The essential difference between the *SW* and other technologies that deal with data such as the *WWW* or databases is that the *SW* is interested in the meaning of data more than in its structure. The *SW* consists of three main technical standards: *Resource Description Framework (RDF)* (W3C, 2004a)*, Web Ontology Language (OWL)* (W3C, 2004b)*, SPARQL Protocol and RDF Query Language (SPARQL).* With the *SW* technologies, *Web 3.0* will be more intelligent, linked, and open. The following section presents a brief description of the Web 3.0.

2.2.1.1 Semantic web standards

In the context of the Web 3.0, the W3C ("Semantic Web - W3C," 2020)has standardized the *SW*. These standards fall in one of the following categories:

- *Linked Data* is about publishing and connecting structured data at Web scale, using technologies like *RDF, RDF* in attributes (*RDFa*), etc.
- *Vocabularies* define concepts/relationships, describe and represents specific domain knowledge using technologies like *RDF schema (RDFS), OWL, etc.*
- *Query* is about retrieving information from the *Web of Data* through *SPARQL*.
- *Inference* allows discovering new relationships on the *SW* through technologies like *Rule Interchange Format* (*RIF*).

The following *SW Stack* illustrates the architecture of the *SW* ("OWL Web Ontology Language Overview," 2020).

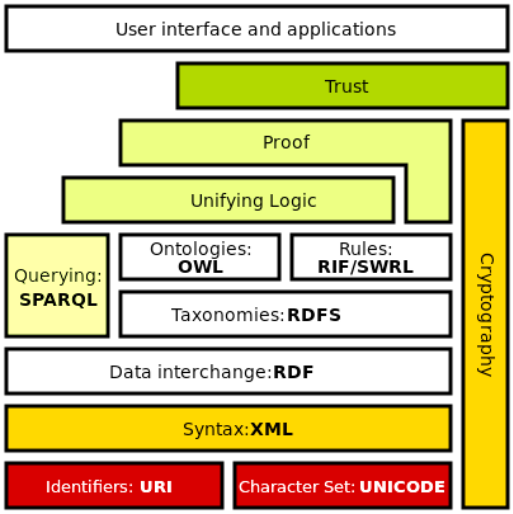


Figure 2.1 The semantic web stack[1]

***URI*:** A Uniform Resource Identifier is a sequence of characters that identifies a specific resource such as a book, document, or page e.g. (https://www.exa.com/rf/rfile.txt). *Uniform Resource Locators (URL)* are a subset of *URIs*.

***XML*:** The *eXtensible Markup Language* defines rules for encoding documents that are machine-readable and human-readable.

***RDF*:** It is an essential standard of the *SW*. It is a language for expressing data models and describe objects and their relationships. *RDF* models can be represented in multiply concrete syntaxes, such as *RDF/XML, Turtle*, *N-Triples, JSON-LD*, etc.

RDF Schema: It is built on top of *RDF* that defines *RDF* vocabularies, classes, property hierarchies, and triples relationship.

***OWL*:** It is a family of knowledge representation that extends the expressiveness of existing standards such as *XML*, *RDF*, and RDFS. It is the ontology language of the *SW,* which represents richer and complex knowledge about things. *OWL* is a fast and flexible data modeling with efficient automated reasoning. *OWL* consists of well-described syntax, well-described semantic, effective reasoner, etc.("OWL Web Ontology Language Overview," 2020). OWL may be classified into three sub-languages: OWL-Full, OWL-DL, and OWL-Lite. The OWL-Lite is the least expressive specie that is aimed for easy application. This sub-language offers a functional subset for the users that help them use the OWL. OWL-Full is the most expressive specie that offers features to be used by knowledge-based systems and relational databases. The third specie is the OWL-DL that its expressiveness falls between that of OWL-Full and OWL-Lite. OWL DL

[1] https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

(Description Logic) is aimed to support the current DL business section and offer a language that has the necessary properties for reasoning. RDF documents will usually be in OWL Full.

SPARQL: It is the *RDF* semantic query language. Similar to *SQL,* uses SELECT, WHERE, etc.

RIF: the *Rule Interchange Format* is an XML language for expressing and interchanging Web rules.

More details about the SW are presented in Appendix A.

By discussing the SW technology, the ontology concept comes into the picture. In our work, the ontology is integrated as an essential feature into the proposed hybrid recommender system in order to increase the accuracy of recommendations. The ontology is integrated as a model to represent the knowledge of the higher education and school domains, career domains, high school student's profile, and university graduates' cases. In the next section the ontology in presented.

2.2.2 Ontologies

In 1993, (Gruber, 1993) has defined the concept of ontology as an "explicit specification of a conceptualization". An ontology is a formal description of knowledge as a collection of classes within a specific domain and the relations that hold between them (Staab and Studer, 2004). Recently, ontologies have attracted extensive attention in designing domain knowledge of courses, e-learning, news, software engineering, etc. The main ontology components are classes (sets of objects), instances (objects/individuals), properties (binary relations between objects), and axioms (semantic constraints on the former). These components are used to design the object-oriented model for specific domain knowledge and share it for reuse on the Web. Moreover, an ontology enables Web content to be understandable by humans and machines alike.

The benefit of using ontologies is that, by describing the relations between concepts constructed into them, they allow automated reasoning about knowledge. Ontologies also offer the value that they are agnostic to data formats such as structured, unstructured, or semi-structured data. This makes them usable to support data integration, data analysis, and concept and text mining. The following figure illustrates the relationship of an ontology example:

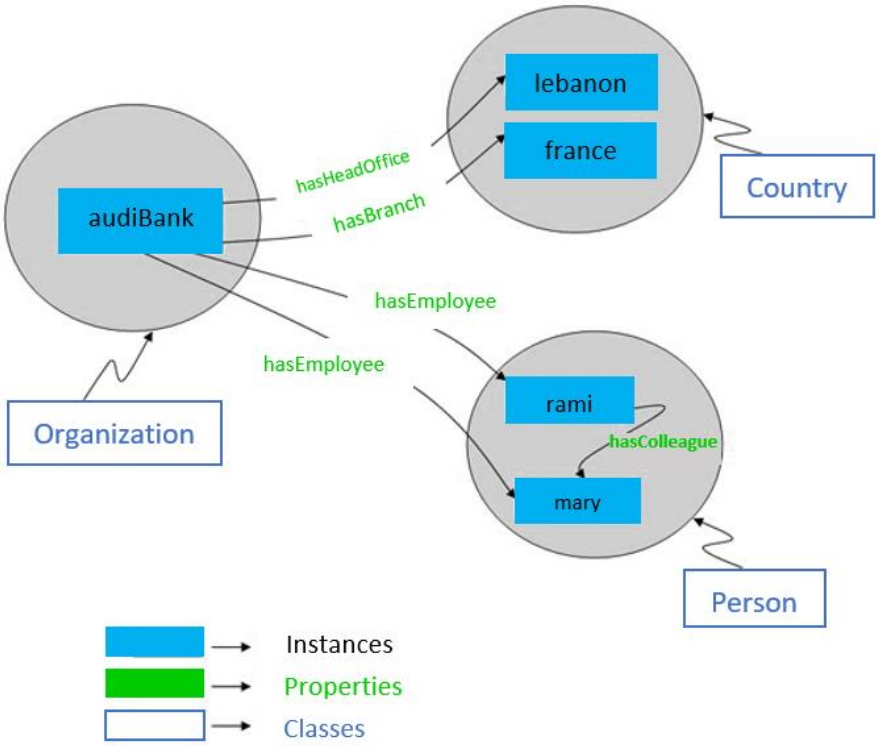


Figure 2.2: Ontology example

Figure 2.2 describes an individual named *rami* as an instance of the class *Person*, and use the property hasEmployer to connect *rami* to the individual *audiBank*, indicating that *rami* is an employee of *Audi Bank*. In the next section, the *OWL* features and syntax are presented.

2.2.2.1 The main roles of ontology

Ontology has a vital role in sharing and representing knowledge in a format that can be readable by machines. The basic contributions of ontology are described as follows (Bürger and Simperl, 2008):

- ***In communication:*** ontology is used for the communication between computational systems, humans, between themselves and between each others.
- ***In computation inference:*** an ontology can be used internally for representing and manipulating plans and planning information. Also, it is used for analyzing algorithms, internal structures, and inputs/outputs of the systems.
- ***In reuse of knowledge:*** an ontology is used for indexing/structuring libraries and repositories. Additionally, ontology can semantically enhance the usage of knowledge (Gündüz-Ögüdücü, 2010) by integrating it into knowledge-based systems such as *Recommender Systems.*

A significant part of our study focuses on the integration of ontologies into a novel hybrid RS approach to increase the accuracy of recommendations. In this hybrid system, the ontology is used as a model in order to represent the knowledge of the higher education and school domains, career domains, high school student's profile, and university graduates' cases. In chapters 5 and 6, details about the integration of ontology into the proposed RS system are presented.

2.2.2.2 Ontology structure

Ontologies can be categorized according to their formality, which determines the degree of the axiomatization of logical instructions. Many approaches in the *SW* have been used for domain modeling such as Thesaurus, Taxonomy, Conceptual models etc.

***Thesaurus*:** this model is used to organize terms of specific domain knowledge with restrictions to lexical relationships such as the homonym and synonym. A well-known example of a *Thesaurus* model is *WordNet*.

Taxonomy: this model represents the formal structure of classes or types of objects within a domain knowledge. Taxonomy is a method of categorizing vocabularies terms into a hierarchical structure. The root of the hierarchical model is the general concept of the tree. The tree nodes represent the terms with a connection to other nodes via parent/child relationships (Boyce and Pahl, 2007). Therefore, machines learn efficiently using taxonomies and can make statistical inferences, statistical associations based on proximity.

Conceptual models: These models are used to express the data structure of domain knowledge by means of classes, attributes, and relationships such as *Unified Modeling Language* (UML) and *Entity Relationship Diagram* (ERD).

The ontology modeling in this thesis is based on the conceptual model approach. The ontology is described in details in the chapter 5.

Although the *SW* technologies are promising, however, many issues, limitations, and challenges are facing it such as semantic interconnection and reasoning, theoretical barriers, and technical obstacles. More details about the ontology are presented in Appendix A.

In the following section, the Case-based reasoning systems are presented. Our proposed hybrid RS integrates the CBR as core technique to retrieve the most similar cases between the high school students' cases and the university graduates' cases.

2.2.3 Case-based reasoning systems

Case-based reasoning (CBR) (Perner, 2019) is an *Artificial Intelligence* (AI) technique applicable to problem-solving and learning where earlier cases are available. CBR is the process of addressing a new problem based on the solutions of similar prior problems. Solutions for the current problem are retrieved from a library of prior cases called case-base. Researchers revealed that most people

collect solutions based on previous experiences with similar cases. For example, a vehicle mechanic who repairs a car engine using his experience with another car that showed similar symptoms is implementing CBR.

The automated reasoning technique (Das et al., 2021) of the CBR defines the problem for the current case, searches the case-base in order to retrieve the most similar prior cases, uses the solutions of the retrieved cases and adapts it to the current case problem, and finally updates the case-base by saving the new experience. Thus, CBR performs as memory and recall is done based on the similarity retrieval and reuse of the most similar solutions. Current addressed problems may be retained and the memory grows as problem-solving happens. Besides, CBR enables the utilization of specific knowledge of prior experienced real cases rather than using only the general knowledge of a problem domain.

Attaining problem-solving, planning, memory, and learning, CBR offers a basis for innovative technology of advanced computer systems that can resolve situations and adapt to new cases. The CBR has the ability to deal with complex real world cases. Therefore, it has been applied in many domains such as recommender systems, e-learning, knowledge management, medical diagnosis, etc.

2.2.3.1 The generic CBR cycle

The CBR system is composed of four consecutive processes. Figure 2.3 illustrates a high-level description of a standard CBR cycle (Perner, 2019) and we describe each of its processes below.

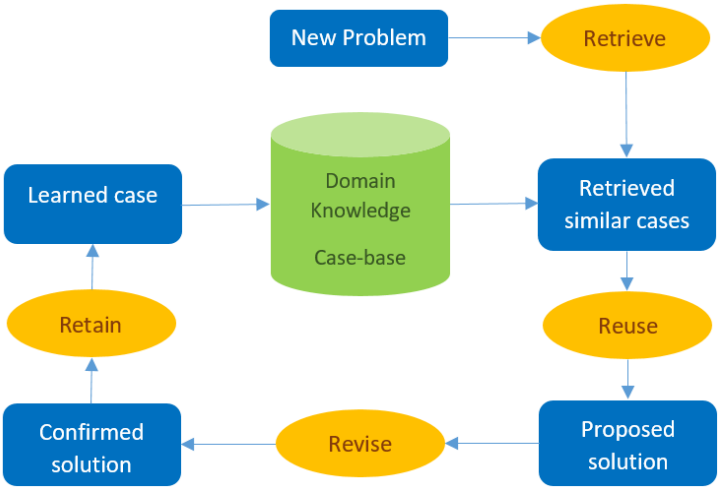


Figure 2.3: The generic CBR cycle

- ***Retrieve:*** the retrieval process begins with a new case or problem and finishes when retrieving the best matching set of prior cases. For this phase the degree of similarity between the cases are defined. Here, the global similarity (Kang et al., 2011) is computed from a weighted assessment of the several feature similarities between cases. The set of

best matches is selected when the similarities between the new problem and the prior problems stored in the case-base are computed.

- ***Reuse:*** the main role of this process is to adapt the prior solutions of the retrieved cases to the new case based on the dissimilarities between them. For example, a RS can identify the active user's interests in similar products through the interest features. Supposing that the user's interest in a new product is same to the user's interest in same products, the RS computes the confidence value of interest for the new product. The confidence value is used to choose whether to suggest the new product to the active user.

- ***Revise:*** in this process, the case's solution produced by the *reuse* process is evaluated. Successful results are learned and the unsuccessful results are repaired by the system using the domain knowledge and/or user feedback.

- ***Learn:*** in this process, the new case is stored into the case-base attached with the interest features that were added in the *revise* process. The case can be updated by adding the user's explicit and implicit feedback about the proposed solution in the case-base.

The CBR cases of the case-base represent several different sorts of knowledge and can be stored in many different formats.

CBR systems has many advantages that differentiate it from other smart systems. CBR performs the simple computation in its *Retrieval stage* on searching for the relevant prior cases. Whereas, the rest of the CBR tasks involve storing and presenting data. Additionally, the CBR permits revision for the retrieved solutions, which allows updating the case-base. More details about the CBR are presented in Appendix A.

In our work, CBR enabled us to utilize specific knowledge of prior experienced real university graduates cases rather than using only the general knowledge of a problem domain. In the next section, the concept lattice and FCA are presented. In our work, FCA is used to analyze university graduates' data such as their interests and preferences during their years of study at the high school, their university majors, and their careers. Additionally, the university graduates trajectories are clustered through the implementation of this technique.

2.2.4 The concept lattice and FCA

Hao (Hao et al., 2018a) introduced the Formal Concept Analysis as a mathematical concept. FCA is a significant technique within the information retrieval domain and concept formalization. This technique is implemented in several fields specifically in machine learning, data mining, data preprocessing, and ontology construction.

FCA data are represented in a matrix where objects' data is illustrated in the matrix rows and attributes' data is illustrated in the matrix columns. The FCA matrix integrates only Boolean values of 0 or 1. This technique is based on the definition of a concept consisting of intension and extension. The concept's objects are included in the extension while the attributes are included in the intension that is present in those objects.

For further information, we refer to ("Conceptual Design of Document NoSQL Database with Formal Concept Analysis," 2016). FCA uses a triple K= (G, M, I) as a formal context. This triple composed of two sets (G, M) and a binary relation (I) between G and M. Components of G are named objects while components of M are named attributes of the context. The fact (g, m) ∈ I means that the object g has the attribute m. A formal context K may be seen as a Boolean matrix relating objects and their attributes.

For any g ∈ G, H ⊆ G, m ∈ M, N ⊆ G, we note:

- $I(g) \stackrel{def}{=} \{ n \mid (g, n) \in I\}$

- $I(m) \stackrel{def}{=} \{ h \mid (h, m) \in I\}$

- $I(H) \stackrel{def}{=} \bigcup_{h \in H} I(h)$

- $I(N) \stackrel{def}{=} \bigcup_{n \in N} I(n)$

	T1	***T2***	***T3***	***T4***
O1	X			X
O2		X	X	
O3	X	X	X	
O4				X
O5			X	

Table 2-1: Context example

As a running example, we introduce the formal context $K0 = (G_0, M_0, I_0)$ where G_0 = {O1, O2, …, O5}, M_0 = {T1, T2, ..., T4}, and each element of I_0 is shown as an X in Table 1. For example, the object O2 has attributes T2 and T3.

- ***The concepts of a context:***

Formal concepts are produced by arranging all objects, which share a set of attributes. In Table 4.1, the cells highlighted in gray show that O2 and O3 share the same attributes T2 and T3, and only those objects share those attributes.

More formally, a formal concept C from a formal context K= (G, M, I) is defined by:

- a subset H of G, named the extension of the concept

- a subset N of M, named the intension of the concept

- I(H) = N and I(N) = H

- ***Calculating formal concepts***

a formal concept is derived from a formal context using a derivation operator shown in the following procedure:

- Choose an object set A as a "candidate" extension.
  A= {O2}
- Derive the intention A'= I (A).
  A'= {T2, T3}
- Derive the maximum extension (A')'= I (A').
  (A')'= {T2, T3}'= {O2, O3}
- (A", A') is a formal concept.
  (A", A') = ({O2, O3}, {T2, T3})

The formal concepts of a formal context K can be organized in a lattice, inducted by the inclusion of their extensions (resp. intentions). The following diagram illustrates the concept lattice of our running example:

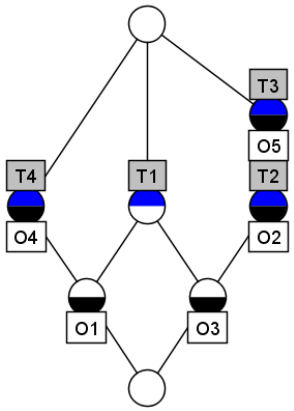


Figure 2.4: Conceptual hierarchy of concepts

Each object *Oi* belongs to the extension of the concept on which it appears, as well as all its descendants in the lattice. Each attribute T*i* belongs to the intension of the concept on which it appears, as well as all its ancestors in the lattice.

- **Transformation of data into a formal context**

The process of data transformation into a formal context is called scaling. The scaling process transforms a many-valued context into a formal context. The following table shows a many-valued context example, where the object "P1" has attributes "Gender = m" and "Age =21" and object "P3" has attributes "Gender (not indicated)" and "Age =66".

	Gender	*Age*
P1	M	21
P2	F	50
P3	/	66
P4	F	88
P5	F	17
P6	M	/
P7	M	90
P8	M	50

Table 2-2: Many-valued context

Missing values in the table are indicated by "/". The many-valued context transformation into a formal context is shown in Table 2-3.

	Gender		***Age***				
	M	F	<18	<40	≤ 65	>65	≥80
P1	X			X	X		
P2		X			X		
P3						X	
P4		X				X	X
P5		X	X	X	X		
P6	X						
P7	X					X	X
P8	X				X		

Table 2-3: Many-valued context transformation

Discrete values (such as "gender" in our example) are replaced with one column per value. Numeric values (such as "age" in our example) are first discretized into a finite number of intervals, and then treated as discrete values.

The concept lattice is represented graphically in Figure 2.5 as a line diagram to support analysis, mining, visualization, etc.

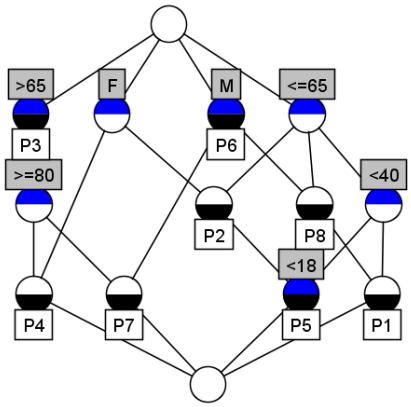


Figure 2.5: Age and gender line diagram

This research focuses essentially on proposing a novel hybrid recommender system in order to recommend personalized recommendations to high school students based on their interests and domain knowledge. Therefore, in the next section, details information about the recommender systems are presented.

2.2.5 Recommender systems

Recommender systems (Aggarwal, 2016a) are the most significant machine learning tools, which predict users' behaviors and recommend personalized items such as *courses, articles, books, movies, playlists, etc.* The origin of RS is based on information retrieval. In addition, this type of systems involves different intelligent techniques such as text analysis, text mining, semantic technologies, artificial intelligence, machine learning, etc.

Since 1990, recommender systems increase in popularity due to the fast growth of internet technology. Recommender systems' purpose is to speed up searches and simplify users' access to items they are interested in. Thus, these systems are becoming more and more vital in internet applications, such as e-learning, e-business, e-service, and e-government. For example, e-commerce companies use recommender systems in order to increase sales and enhance customers' experience. Smart recommender systems help users to make the right decisions by quickly finding their required items or services.

The object that a RS analyzes and suggests is named an item. Every item has properties that are called features or attributes. For example, a university is an item in a university RS and can be described by attributes such as university location, university name, university fees, etc. The activity that a student submits his/her interests of an item is named rating. A student's rating history includes all prior ratings he/she submitted. For example, a student's rating history involves eight ratings. His/her rating profile consists of three ratings on universities, three ratings on vocational

colleges, and two ratings on e-learning systems. The return of a RS includes item recommendations to a user and generating personalized item information. However, a RS does not always generate a list of recommendations; it can also guide users to find items that are probably interested in. For example, it can displays popups or advertisements on interesting products.

RS can be implemented in many applications areas such as entertainment, e-commerce, e-services, social network, content applications, etc.

Entertainment Applications: Movies, music, television programs are some examples of areas that use recommender systems. *Netflix* and *YouTube* are the most two known examples.
E-commerce Applications: *E-bay* and *Amazon* are the most popular e-commerce sites that use recommender systems to generate suggestions to their customers.
E-services applications: services such as hotel and travel reservation use recommender systems to generate recommendations to active customers. Expedia and TripAdvisor are two popular Web applications that rely on recommender systems to provide personalized suggestions to their active users.
Social Network Applications: *Twitter, LinkedIn,* and *Facebook* use recommender systems to recommend friends, pages, products, etc.
***Content Applications*:** such as articles, documents, news uses recommender systems. This type of application provides recommendations based on users' previously read articles or content. CNN and BBC are two important examples that use content technique.

Recommender systems collect several types of information about users' interests, preferences, tastes, etc. Two categories of information are integrated into recommender systems in order to suggest adequate recommendations.

***First category*:** This is the characteristic information about items such as (keywords and categories) and users such as (preferences, interests, and profiles).

Second category: This is the user-item interaction information such as (ratings, likes, and total of purchases).

Every RS employs a filtering technique to retrieve the needed suggestions and items for the users. Several filtering techniques have been employed to the basic challenge of providing accurate and efficient recommender engines. These techniques are mainly categorized into various types described as follows:

- *Collaborative filtering* recommender systems (CF), which are based on user-item interactions.
- *Content-based* recommender systems (CB), which are based on characteristic information.
- *Knowledge-based* recommender systems (KB), which are based on domain knowledge.

- *Demographic-based* recommender systems (DF), which are based on users' demographic data.
- *Hybrid* recommender systems, which are based on mixed types of information in order to overcome the limitations of the CF, CB and DF systems.

The data required for recommender systems are acquired in two ways (implicitly or explicitly):

- Explicit data are acquired from user ratings after listening to a song or watching a movie.
- Implicit data are acquired form purchase history, search engine searches, or users/items' knowledge.

Figure 2.6 illustrates the required data for the basic recommender system techniques.

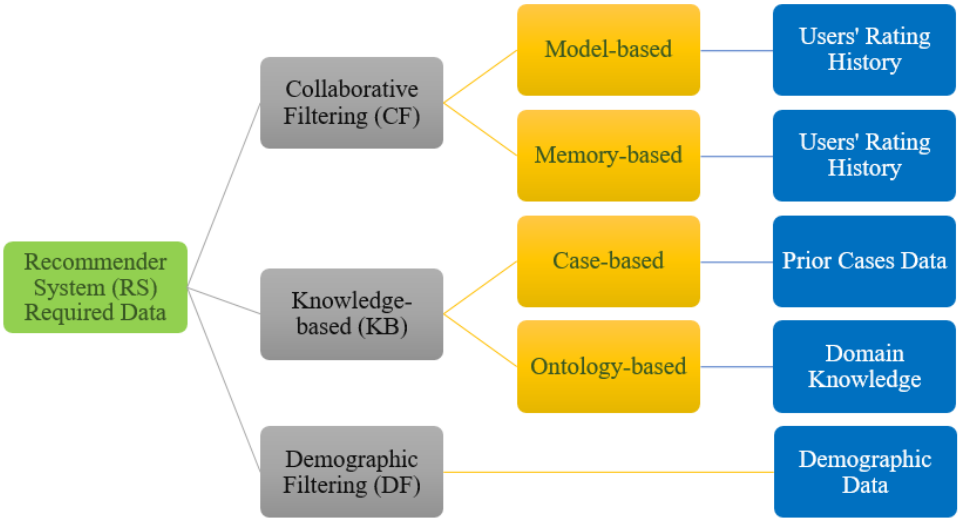


Figure 2.6: Required data for recommender systems

In the next section, the basic types of recommender system techniques are presented.

2.2.5.1 Basic types of recommender system techniques

We can distinguish between many techniques used in recommender systems. Recommendation techniques such as Demographic-based (Liu and Callvik, 2017), Knowledge-based (Burke, 2000) (Constraint-based (Burke, 2007), Case-based reasoning(Szczepaniak and Duraj, 2018), Ontology-based (Bahramian and Abbaspour, 2015)), Content-based filtering (Aggarwal, 2016b), Collaborative Filtering (Cheng et al., 2016) (Memory-based (Ghazarian and Nematbakhsh, 2015), Model-based (Do, 2010)), and hybrid recommender systems (Burke, 2007) are widely used in several domains. The following figure illustrates the taxonomy of the most popular recommender systems techniques.

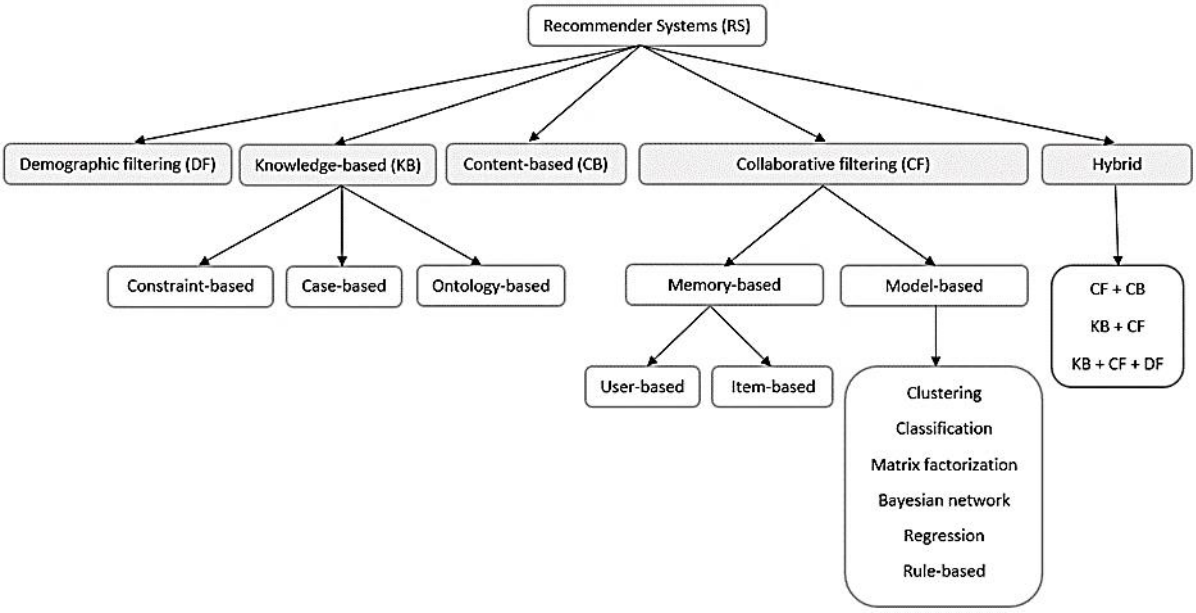


Figure 2.7: Recommender system techniques

2.2.5.1.1 Collaborative filtering RS

The CF recommender system is the most popular and successful approach implemented in the recommendation domain. This technique is based on the notion that if some users have the same preferences in their history, they will share mutual preferences in the future (Cheng et al., 2016). The CF technique integrates users' preferences, interests, and actions to suggest products to users based on the match between users' profiles (Zarzour et al., 2018). The following figure illustrate the CF technique.

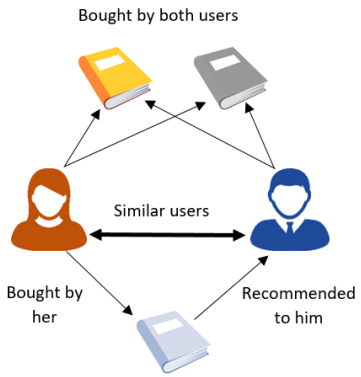


Figure 2.8: CF technique

The CF algorithm encompasses two essential types namely the Model-based and Memory-based techniques.

A. Memory-based

This technique computes the similarity between users based on users' activities, ratings, or selected items to generate appropriate recommendations. Memory-based integrates users and items' dataset to generate predictions. The Memory-based implements two different methods namely the user-based and item-based technique (Ghazarian and Nematbakhsh, 2015). The user-based consists of associating similar users to the active user, recognized as neighbor users. Furthermore, missed ratings are predicted using various similarity metrics. The metrics calculate the similarities values based on past users' ratings. The Item-based technique function is to focus on the items instead of the users. This technique works on finding the most similar items based on the active user's ratings compared to past users' rating history. Also, unknown items' ratings are predicted based on the items' ratings of the active user. CF technique implements several similarity metrics such as the Euclidean Similarity, Pearson Correlation Similarity, Cosine Similarity, and their adjustments to compute similarity values.

B. Model-based

This technique calculates the similarities between users and/or items, then saves them as a model, and then implements the saved similarity values to generate recommendations. The Model-based implements several algorithms such as clustering algorithm, matrix factorizations, Bayesian network or regressions (Do, 2010).

Nevertheless, the CF technique has many problems such as the *cold start* for new users, *scalability*, *sparsity*, grey-sheep, and many limitations such as treating heterogeneous data types and high dimensional datasets (Breese et al., 2013).

- ***The cold start problem***

The CF technique needs past users' history such as users' activities and ratings to generate precise recommendations. The *cold start* problems occur when the dataset does not include sufficient ratings and preferences. Therefore, reliable recommendations will become hard to provide. Usually, the *cold start* issue happens due to three main reasons: new active user, new community or a new item added to the system (Schafer et al., 2007). For example, if a new active user asks for a recommendation, the system will find difficulty to match him to similar users, since not enough history exists about his activities or ratings in the database. Hybrid systems are used here to overcome this problem.

- ***The sparsity problem***

This problem occurs when the items and users matrix table is widely sparse. In this case, the precision of the recommendations will decrease since past users could not rate all the available products in the system. Hybridization is commonly used for improving recommendation techniques and solve the data sparsity issue. For example, combining the CF and DF recommendation techniques is a method to minimize the sparsity problem of the CF algorithm.

- ***The scalability problem***

An enormous community of users and products exists in several of the environments that the CF systems make a recommendation in. Hence, great computation power is necessary to compute recommendations. Dimensionality reduction and clustering techniques are ways to overcome this challenge.

- ***Grey-sheep problem***

This problem is caused by odd recommendations since the user may have other features that do not match with any other user or community of users (de Campos et al., 2010) . An example of a grey-sheep issue is when a user neither agrees nor disagrees with any user or group of users. Grey-sheep issue can increase the error rate in recommendations and can affect the performance precision of the RS. Also, this issue possibly will negatively affect the predictions for the rest of the community in the dataset (Bruke, R., 2002).

- ***Treating heterogeneous data types limitation***

Basic recommender filtering techniques have no capability to treat heterogeneous data types. Here comes the role of the hybrid recommender systems that can handle and compute heterogeneous data.

- **Treating high dimensional *datasets* limitation**

Basic recommender filtering techniques have no capability to deal with high dimensional datasets. High dimensional datasets encompasses high number of attributes. This limitation can be addressed by decreasing the number of attributes in the dataset or using a Hybrid RS that can handle large datasets.

2.2.5.1.2 Content-based RS

The CB technique works with the data provided by the user. Users' data is collected either explicitly by rating or implicitly by clicking a hyperlink. The CB algorithm function is to find products with the same content to suggest to the active users. The CB recommendations are based on what the active user liked. This recommender system compares the user's items ratings with items he or she did not rate and then computes the similarities. Based on that, the recommender system recommends the appropriate items, which are similar to the rated ones (Esfahani and Alhan, 2013). The following figure illustrates the CB method.

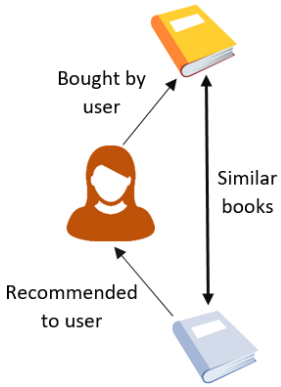


Figure 2.9: Content-based method

This method does not have a cold-start problem since it integrates features of the content like the actors or categories to recommend similar content. It differs from CF, however, by deriving the similarity between items based on their content (e.g. title, year, description).

The following are the limitations of the CB filtering technique (Puntheeranurak and Tsuji, 2007):

- Users have to rate enough number of items in order to help the system to create a user profile interests and enable the recommender system to generates accurate recommendations.
- The CB filtering technique is purely based on content. Therefore, if two different items share common content, the recommender engine is not capable to distinguish between them.
- The TF-IDF technique is not adequate with terms synonyms. For example, the term "vehicle" and "automobile" are not considered the same, while they have the same meaning.

2.2.5.1.3 Knowledge-based RS

The KB recommender system generates appropriate recommendations based on explicit and implicit knowledge about the users and items. This technique integrates knowledge such as users' characteristics, preferences, interests, or needs (Burke, 2000). KB recommender systems deal with the cases in which ratings are not used for the recommendations. Therefore, ontology is essential in the KB system to overcome the cold-start issues. Ontology is a KB technique, which does not take into consideration users and items past information (Shishehchi et al., 2012).

In the domain of e-learning, the KB technique integrates users' and courses' knowledge and implements it in the recommendation process (Shishehchi et al., 2012). KB techniques are good examples to hybrid them with different recommendation techniques such as CF and CB. However, knowledge-based recommenders require knowledge engineering skills like the domain ontology

design (Burke, 2007). The KB RS has three basic recommendation techniques namely the *Constraint-based RS, Cased-Based Reasoning RS,* and *Ontology-based (OB) RS*:

A. *Constraint-based RS*

In this technique, users identify the constraints (e.g., a range greater than and less than) on the item attributes. Additionally, rules are used to associate users' constraints with item attributes. Furthermore, users can modify their original constraints or requirements based on the results. This process could be repeated until the users get appropriate results.

B. *Cased-based reasoning RS*

It is a technique used for mixing problem resolving with case learning. It is a successful machine learning technique applied in the AI field. This technique uses a case base to compute problems submitted by the active users. When the active user asks the CBR system to solve a new case or problem, a solution is generated by comparing the new case with past cases and find the most similar ones. Every successful and unsuccessful solution will be saved in the case base and used for future case solving (Szczepaniak and Duraj, 2018). The CBR cycle is composed of four main processes. The first process is "retrieve"; the retrieval obtains the most similar cases given a query. The second process is reuse; the reuse adjusts the solutions of the acquired cases to fit the query criteria. The third process is "revise"; in revise, the suggested solutions are tested for success and repaired if they fail. Finally, the fourth process is "retain"; in retain new cases will be stored for future problem-solving. By solving and saving new cases, the CBR learns new experiences. The four main components of a CBR case are the problem description, problem solution, results, and solution's justifications. In the CBR technique, cases are identified as targets.

C. *Ontology-based (OB) RS*

Ontology is a formal conceptualization of many domains knowledge such as education, tourism, medicine, etc... The semantic Web uses the ontology as a backbone to represent domain knowledge. The set of concepts and their relationship with domain knowledge are described by the ontology in a machine-understandable language (Bahramian and Abbaspour, 2015). The four essential components of ontology are classes, instances, properties, and rules. Classes have a certain number of features and describe the concepts of a specific domain abstractly. The classes contain instances that have specific significance like age, color, and gender. The semantic relationships are the properties between classes such as 'is-a' which is a property that describes the conceptual model of the class structure. This conceptual model shows the classes and their subclasses' hierarchy. The RS based on ontology showed a significant role in the recommendation technology (Bahramian and Abbaspour, 2015). Moreover, the ontology-based RS is one type of the KB RS. This type of RS uses ontology engineering to model the knowledge of the domain, users, and items. To find the similarity between concepts, the ontology is needed to compute the semantic relationship. Also, the conceptual model of the ontology structure permits the reasoning

at all concept levels. Thus, to overcome the limitations of the traditional RS, the ontology is integrated to represent the domain knowledge and compute the semantic similarity between the concepts.

2.2.5.1.4 Demographic-based RS

DF recommender system purpose is to cluster the users based on their personal features in order to suggest appropriate recommendations. The following figure shows a DF simple example based on demographic data (Liu and Callvik, 2017).

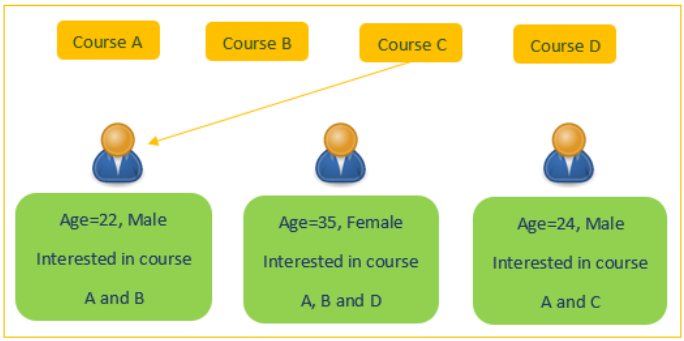


Figure 2.10: DF based on popularity

Figure 2.10 shows a male individual that has an age of 22 years old, and has interest in course A and B and probably has interest in course C. Course C is recommended to this individual based on the most similar demographic data and interests of the individual that has an age of 24 and gender male.

The DF recommendations are based on users' demographic data such as age, gender, education, occupation, address (city, country), etc… The importance of such system is that it overcome the new user issue of the CF technique since they do not require user ratings. Also, it is easy to preprocess the data since it does not require domain knowledge. In DF, it is easy to identify similar users since new user must register and enter his/her demographic data to the system. For example, Table 2-4 shows info on the Age, Gender, Education, City and Native language, etc.

Name	Age	Gender	Education	City	Native Language
Rami	16	Male	Academic	Beirut	Arabic
Mary	17	Female	Academic	Tripoli	Arabic
Steve	14	Male	Vocational	Jbeil	French
Mazen	18	Male	Academic	Jounieh	English

Table 2-4: Example of demographic data

The DF recommender system computes users' similarity and provides personalized recommendations to users taking into account users' demographic data such as gender, age, marital status, location, language, and other personal features (Burke, 2007). This type of data can be

collected through surveys and questionnaires. In DF, users having the same demographic characteristics may also have similar preferences or tastes. Unlike the CF and the CB, the DF technique does not require users' rating history(Liu and Callvik, 2017). The DF technique classifies similar users by computing their demographic data similarities in order to generate accurate recommendations. Also, this type of RS could be combined with the CF or CB technique to improve the accuracy of the system's recommendations (Bruke, R., 2002). The new user's problem does not occur in the DF systems process since they do not require the new user's list of ratings.

The three stages of the DF process are *data input, similarity computation,* and *recommendation computation*. The *data input* stage holds the demographic data of all the users contributing to the system. The *similarity computation* stage employs users' demographic data to associate the most similar users with the active user. Finally, the *recommendation computation* stage generates recommendations to the active user based on the two prior phases.

2.2.5.1.5 Hybrid RS

Several recommendation techniques are being implemented to generate recommendations for active users such as CF, CB, DF, KB, etc. However, these filtering techniques have many limitations and problems. Thus, researchers worked on incorporating more than one technique in a uniform system called hybrid in order to generate precise recommendations. Researchers revealed that hybrid recommender systems can perform better than any standalone filtering technique. Therefore, hybridization methods are commonly used for specific domains in order to improve recommendations' accuracy and overcome the limitations and problems of basic filtering techniques. The following figure illustrates a hybridization strategy example, which combines the CB and CF techniques.

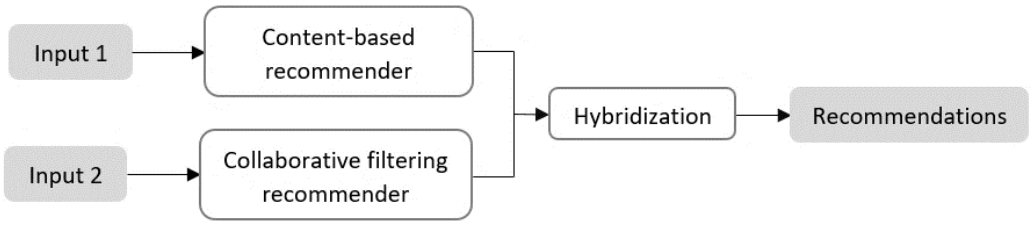


Figure 2.11: A hybridization strategy example

Burke (Bruke, R., 2002) classified the hybrid RS techniques into seven hybridization strategies namely (*weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level)*. The following table describes the seven hybridization strategies:

Hybrid strategy	***Description***

Weighted	The values of two or more RS are collected to generate a single recommendation.
Switching	The system navigates between the hybrid recommendations systems taking into account the running case.
Mixed	The output of two or more recommendation techniques are generated simultaneously. For example, CF rank (3) + CB rank (2) → Combined rank (5).
Feature combination	The features from different sources are integrated into a single RS technique.
Cascade	The running recommendation technique refines the output of a second recommender system.
Feature augmentation	The output of one recommendation technique is integrated as input attributes into a second recommender system.
Meta-level	The model learned by one recommender is integrated as input into another recommender system.

Table 2-5: The seven hybridization strategies

Different hybrid recommendation techniques have been used in different fields. The "feature combination" strategy has been used to combine the CF and CB technique (Vall et al., 2019). Likewise, the "switching" strategy has been used to combine the CB and CF technique to develop a hybrid recommender system (Towle and Quinn, 2000). Moreover, the combination of the CF and KB technique used the "weighted" strategy to enhance recommendations (Billsus and Pazzani, 2000). Besides, the KB and CF implemented the "feature augmentation" strategy to form a hybrid RS in order to achieve more robust recommendations(Kolbert, 2017). Finally, ontology-based and CF have been combined to improve the performance of the CF technique and overcome its limitations (Burke, 2007).

In our approach, we implemented the "feature augmentation" strategy by incorporating the CF and KB techniques in a uniform hybrid RS based on CBR and ontology. More details about our hybrid approach are described in chapter 5 section 5.3.4.

In the next section, the similarity metrics, neighborhood-based CF algorithms, and evaluation metrics are presented. In our work, we used the similarity metrics and neighborhood-based algorithms in order to retrieve the most accurate similarity results for the objects integrated in our study. In addition, the evaluation metrics were used to evaluate the applied similarity metrics and algorithms.

2.2.6 The similarity metrics, neighborhood-based CF algorithms, evaluation metrics, and the evaluation algorithms

Usually, people count on recommendations given by other people that are linked to different domains or products. Thus, recommender systems offer users the capability to count on the preferences or interests of large communities. In order to generate personalized recommendations, a RS makes some similarity evaluations on the users' preferences or interests, and choose which

recommendations match users' tastes. Thus, what is the similarity between two items? In all situations, a full similarity is an absence of differences. Therefore, similarity metrics in a RS are about matching products or users that are most similar. In our study, we aimed to implement and evaluate many RS similarity metrics in order to select the appropriate metric that generates better recommendations to the high school student.

Our hybrid approach incorporates the CF and KB recommender techniques that are supported by the CBR and ontology. This hybrid system is evaluated using two evaluation phases. The first phase evaluates only the CF similarity metrics and neighborhood-based CF algorithms using the evaluation metrics namely the "*Mean Absolute Error (MAE)"* (Bagchi, 2015) and "*Root Mean Squared Error (RMSE)"* (Bagchi, 2015). The second phase evaluates the efficiency of the overall hybrid system using two evaluation algorithms (Recio-García, J. A. et al., 2014) namely the "*Hold Out Evaluator" and "SameSplitEvaluator"*. Thus, the next four sections present the similarity metrics, neighborhood-based model, evaluation metrics, and evaluation algorithms that were applied to the proposed approaches. More details about the implementation of the two evaluation phases are described in chapter 6 sections 6.2.1 and 6.2.5.

2.2.6.1 The similarity metrics for the CF recommender systems

Several similarity measures are available for the *user-based and item-based CF* recommender system approach such as *Euclidean Distance Similarity, Pearson Correlation Coefficient Similarity, Spearman Correlation Coefficient Similarity, City Block Similarity,* and *Uncentered Cosine Similarity.*

The *Euclidean Distance* is the most common among all the distance measures. This distance is a straight-line distance between two vectors. The *EuclideanDistanceSimilarity* technique in mahout java library (Bagchi, 2015) calculates the similarity between two users X and Y. This technique considers items as dimensions and preferences as points along those dimensions. The distance is calculated using all items where both users have a similar preference for that item. It is the square root of the sum of the squares of differences in position along each dimension. The similarity could be computed as 1 / (1 + distance) and the distance is mapped between (0, 1]. The distance between two points with coordinates (x, y) and (a, b) is given by

$$dist((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2} \quad \textit{Equation 2-1}$$

In Euclidean distance, the value of the distance is smaller when users are more similar. The larger the distance value is, the smaller the distance is. Thus, the closer the distance, the greater the similarity (Bagchi, 2015).

The *PearsonCorrelationSimilarity* is based on the Pearson correlation. The values for users A and B are calculated as follows:

- *SumA2:* the sum of the square of all A's preference values.
- *SumB2:* the sum of the square of all B's preference values.
- *sumAB:* the sum of the product of A and B's preference value for all items for which both A and B express a preference.

To calculate the correlation the following formula is used: sumAB / sqrt(sumA2 * sumB2).

$$sim(a,b) = \frac{\sum_p \in P(r_{a,p} - \overline{r_a})(r_{b,p} - \overline{r_b})}{\sqrt{\sum_p \in P(r_{a,p} - \overline{r_a})^2}\sqrt{\sum_p \in P(r_{b,p} - \overline{r_b})^2}}$$

Equation 2-2

a and b represents two users or items, p represents an item, $r_{a,p}$ and $r_{b,p}$ represent the user ratings from a and b for p, and average ratings of r_a and r_b are, for the item or user a and b cite. Here the Pearson correlation coefficient is equal to the covariance of the two variables divided by the standard deviation of the two variables. The results range between [-1, 1], the larger the absolute value, the stronger the correlation, and the negative correlation has little significance for the recommendation.

The *SpearmanCorrelationSimilarity* is like the *PearsonCorrelationSimilarity*. However, the *SpearmanCorrelation* compares the relative ranking of preference values instead of preference values themselves. Each user's preferences are sorted and then assigned a rank as their preference value, with 1 being assigned to the least preferred item. The equation for Spearman Correlation Similarity is given below:

$$w(a,b) = \frac{\sum_{i=1}^{n}(rank_{a,i} - \overline{rank_a})(rank_{b,i} - \overline{rank_b})}{\sigma a * \sigma b}$$

Equation 2-3

The calculation in Spearman Correlation Similarity is very slow and there is a lot of sorting. Its results range between [-1.0, 1.0], 1.0 when there is a total match, -1.0 when there is no match.

The *City block distance* (Giacomelli, 2013) also referred to as Manhattan distance. It calculates the distance between two points, a and b, with k dimensions. The City block distance is computed like following:

$$\sum_{i=1}^{n} |a_i - b_i|$$

Equation 2-4

The *City block distance* result should be greater than or equal to 0. The result for identical points should be equal to 0 and greater than 0 for the points that express little similarity.

The *UncenteredCosineSimilarity* is an implementation of cosine similarity. Its result is the cosine of the angle formed between two vectors. The correlation between two points, *a* and *b*, with *k* dimensions is computed as:

$$Similarity = \frac{\sum_{i=1}^{n} a_i * b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} * \sqrt{\sum_{i=1}^{n} b_i^2}}$$ *Equation 2-5*

This correlation ranges from (+1 to -1). The highest correlation is equal to +1 and the dissimilar points have a correlation equal to -1.

In chapter 6, all the above similarity metrics were applied to our proposed user-based and item-based CF approach. The neighborhood-based CF algorithms also were implemented and evaluated in chapter 6 section 6.2.1. The following section presents the neighborhood-based CF algorithms.

2.2.6.2 The neighborhood-based CF algorithms

The two types of neighborhood-based CF algorithms are the User-based CF and Item-based CF. The difference between the User-based CF and the Item-based CF is that User-based takes the rows of ratings matrix and Item-based takes the columns of ratings matrix for similarity measurement.

In *User-based*, the item's recommendation rating for a user is calculated depending on those items' ratings by other similar users. The ratings are predicted using the ratings of neighboring users. In User-based, the Neighborhoods are defined by similarities among users.

In *Item-based*, the item's rating is predicted based on how similar items have been rated by that user. The ratings are predicted using the user's own ratings on neighboring items. In Item-based, the Neighborhoods are defined by similarities among items.

In our study, we implemented and evaluated the two neighborhood-based CF algorithms. The two neighborhood algorithms were tested using many similarity metrics such as the Euclidean distance, Pearson Correlation, Spearman Correlation, etc. The results of our experiments showed that the User-based CF algorithm generated better recommendations than the Item-based CF algorithm based on our courses' ratings dataset. More detail about the experiments are presented in chapter 6 section 6.2.1. The following section presents the evaluation metrics such as the MAE and the RMSE that are used to evaluate the recommender systems accuracy.

2.2.6.3 The evaluation metrics

Many researchers found several evaluation metrics to evaluate the quality of the prediction. Prediction accuracy metrics find values that show how much the prediction is close to the real preference. The evaluation metrics help to assess the precision of the RS recommendations by comparing the predicted ratings with the rating of the active user. There are many prediction accuracy metrics used for testing the prediction accuracy of the used algorithms such as the Mean Absolute Error (Bagchi, 2015) and Root Mean Squared Error (Bagchi, 2015). In our graduates' dataset context, MAE and RMSE will assess how well the RS can predict a user's rating for a course/career.

The MAE metric evaluates the accuracy of an algorithm by comparing the value of predictions against the actual user's ratings for the user-item pairs in the test dataset. For each rating prediction pair, their absolute error is calculated. After summing up these pairs and dividing them by the total number of rating-prediction pairs, Mean Absolute Error can be found. It is the most commonly used and can be interpreted easily. The equation of Mean Absolute Error is given in below.

$$MAE = \frac{\sum_i^n |r_i - e_i|}{n}$$ *Equation 2-6*

The RMSE is calculated by finding the square root of the average squared deviations of a user's estimated rating and actual rating. Once rating-prediction difference is calculated, the power of 2 is taken. After summing them up and dividing them by the total number of rating-prediction pairs and taking square root of it, Root Mean Square Error can be found. The equation of Root Mean Square Error is given below.

$$RMSE = \sqrt{\frac{\sum_i^n (r_i - e_i)^2}{n}}$$ *Equation 2-7*

Where in both formulas for MAE and RMSE n is the total number of items, i is the current item, r_i is the actual rating a user expressed for i, and e_i is the RS's estimated rating a user has for i. The smaller RMSE and MAE are, the more accurate a RS. This is because RMSE and MAE will calculate smaller values if the deviations between actual and predicted ratings are smaller. By using evaluation metrics, prediction accuracy and efficiency of the CF methods can be calculated and compared.

In our study, the MAE and RMSE are used to evaluate the accuracy or our proposed recommender systems. The next section presents the evaluation algorithms that are used to evaluate the overall accuracy of the KB recommender systems' approaches.

2.2.6.4 The evaluation algorithms

To evaluate the accuracy of our recommender system, two jColibri evaluation algorithms (Recio-García, J. A. et al., 2014) were implemented namely the *HoldOutEvaluator and SameSplitEvaluator.*

- The *HoldOutEvaluator* method splits the case-base into two sets, one used for testing where each case is used as a query, and another that acts as a normal case-base. This process is performed several times.
- The *SameSplitEvaluator* method splits the case-base into two sets, one used for testing where each case is used as a query, and another that acts as a normal case-base. This method is different from the other evaluator because the split is stored in a file that can be used in following evaluations. This way, the same set is used as queries for each evaluation. The generateSplit() method does the initial random split and saves the query set in a file. Later, the HoldOutfromFile() method uses that file to load the queries set and perform the evaluation.

These two evaluation algorithms helped us to evaluate the accuracy of our proposed hybrid recommender system that is supported by the ontology and CBR system. In the following section, the related works of the FCA and recommender systems are presented.

2.3 Related works

In the next section, the FCA related works are presented.

2.3.1 *FCA related works*

FCA is implemented in many fields such as knowledge mining, text mining and machine learning. In this section, we describe researches that focused on pattern and concept discovery based on the FCA clustering technique. (Jay et al., 2013) proposed an approach based on FCA to explore trajectories of care. This method clustered care trajectories for breast cancer. The authors gathered data about all admissions in the country from the French national case mix system. They focused on breast cancer patients who had undergone surgery and they recomposed their trajectory of care occurring after surgery. The authors used analyses of hospitalizations to produce illness profiles and computed cumulative hospital costs for each patient. In this approach, FCA generated an automatic classification of care trajectories that can be used to setup cost-of-illness studies.

In the same context, authors in this paper (Egho et al., 2011) presented a different approach by combining two methods sequential pattern mining and formal concept analysis. They applied these two techniques on real medical applications to mine patterns of trajectories of care in a French medico-economic dataset. Results showed the importance of concept lattice properties and the ability of this approach to classify and discover interesting sequential patterns. However, this study needs further development on several axes.

In addition to the two approaches above, authors in this study (Poelmans et al., 2013) discussed the potential of FCA to analyze unstructured police reports. The authors presented a knowledge mining approach with acceptable results obtained through empirical validation. In the aforementioned paper, experiments were based on three case studies. The first case study worked on allocating domestic violence from 4,814 reports. The second case study focused on identifying and investigating human trafficking suspects acquired from 266,157 reports. The third case focused on identifying radicalizing subjects from 166,577 police reports. FCA discovered confusing situations, anomalies, new concepts, and faulty labeled cases. Moreover, FCA proved its effectiveness in profiling suspects and their evolution over time.

Studies showed that the FCA tool is an effective technique for social network processing and analysis. In this paper (Silva et al., 2017), the authors discussed an FCA method to discover professional behavior through data extracted from the LinkedIn social site. In this work, the authors used FCA to identify relations among professional competences. Additionally, results showed the minimum sets of skills that are required to get the job position.

Moreover, in this research (Bal et al., 2011) the FCA method was used in an experimental study on employee recruitment to model the qualifications of applicants during the recruitment process by taking into consideration the required qualifications for the job position. The concept lattice modeled the qualifications of the candidates who applied for a specific position. Besides, the implications and the association rules were computed to help the decision-makers select the appropriate candidate for the available position.

Furthermore, experiments show the effectiveness of the FCA technique in many domains especially in road safety. For instance, in this paper (Allani et al., 2018) authors presented a new geocast technique named Data Dissemination Protocol based on Map Splitting (DPMS). The main function of DPMS is mining associations between vehicle trajectories and crossed regions to build the zones of relevance. Authors relied on the Formal Concept Analysis to extract significant clusters from relational data. Results showed the effectiveness and efficiency of the new DPMS approach that outperformed its competitors.

Additionally, in this paper (Hao et al., 2018b) authors used the *FCA* technique to discover human actions from non-intrusive sensor data. They implemented *FCA* to study the problem of multi-resident action recognition and identify the relations among sequential behavioral patterns. Activity recognition is an essential part of smart home applications. This approach outperforms the CASAS multi-resident benchmark database. Accordingly, the authors presented a technique of sequential pattern mining to identify the ontological features of sensor data.

A more advanced assessment structure was proposed in (Hans, 2016). The authors proposed a new technique, which incorporates the student evaluation table with the concepts acquired from the table of skill questions. The table includes the objects as test questions and the attributes as the learning skills that are required to respond to the questions. This table represents the formal context

and the applied learning skills were of type numerical, analytical, mathematical, and linguistic. The concept lattice is should be created and the context should be readied before the evaluation. Here, the performance parameters represent the obtained concepts that are included in the assessment process specifying the student adept in the skills. This approach emphasizes the student knowledge and level of different learning skills that provide a more improved evaluation.

In this study (Škopljanac-Mačina and Blašković, 2014), the authors discussed a sample of a physics exam at the university. The objects are represented by the tasks of the exam and the attributes are represented by the physical concepts required to resolve the tasks. The sorting algorithm of topology is applied to the concepts after the generation of the concept lattice of the context. Also, the optimal order of the exam questions can be selected using the obtained concept lists. Therefore, questioning in the correct order is particularly significant in e-learning courses where test questions are generated automatically.

In all the above studies and approaches, *FCA* demonstrated its effectiveness in analyzing and grouping data in different domains such as care, recruitment, police investigations, education, transport efficiency, social network and activity recognition. Our study aims to apply the *FCA* method in the educational domain to analyze and cluster students' trajectories.

2.3.2 Recommender systems' related works

Researchers have proposed several approaches for building recommender systems, which offer recommendations to users based on specific criteria that match their interests. However, all these recommendations have strengths and weaknesses, which makes the prediction process to fit a specific domain and dataset complexity. In the next section, the related works of the Hybrid recommender systems are presented.

2.3.2.1 Hybrid recommender systems

Hybrid systems have been a widely held research field and have shown improved returns than using any standalone filtering technique. The following are some researches that discuss the implementation of hybrid recommender systems in several domains.

Several hybrid combinations between CF and DF techniques have been proposed in many studies and researches. This type of hybridization can minimize the limitations of the CF technique such as the sparsity issue. In addition, this technique can overcome the new user problem because the DF technique does not need the user's rating history.

Many researches have been conducted used this technique as follows: Researches worked on finding the similarities between the users based on similarities in their profiles and characteristics. Rather than solving the "cold-start problem", researchers proposed the collaborative filtering and demographic-based approaches of hybridization to improve the movies recommendation quality (Schafer et al., 2007). In addition, this paper (Xia et al., 2009) presented an augmentation item-

based CF hybrid system using demographic data to predict missed data such as age and occupation information. (Vozalis and Margaritis, 2007) used demographic data and rating history and integrated it in a feature combination hybrid RS to enhance the item-based CF technique for movies recommendations. The same researchers, (Vozalis and Margaritis, 2007) presented a hybrid RS, which mixes CF and SVD techniques based on an augmenting approach. This system also used demographic data to enhance the accuracy movies recommendations. Additionally, in movie domain, researchers (Ruchika et al., 2015) presented a hybrid RS based on users' demographic data to improve the system's predictions. Here, the authors categorized the movie types based on users' demographics attributes such as age, gender, if he/she has children, and if he/she is a student (yes or no). Furthermore, in this paper (Agarwal et al., 2017), the authors used users' demographic data instead of users' rating history to generate accurate movies recommendations and overcome the CF cold-start issue. Additionally, they presented a metric tool to evaluate the efficiency of using demographic data in such a system.

Eventually, few hybridization approaches have been proposed with a combination of three filtering techniques such as CF, KB, and DF. This hybridization approach is not well known. This paper (Benouaret, 2017) proposed a hybridization strategy consisting of three core techniques namely the demographic, semantic and CF. The goal of this RS is to enhance the visitor's experience in visiting museums and tourist places. Each method is adapted to a specific stage of the museum visit. The demographic approach is first used to overcome the CF cold-start issue. The semantic approach is then activated to provide recommendations to the user semantically close to those he/she has previously appreciated. Finally, the collaborative approach is used to recommend to active user works previously liked by similar users.

2.3.2.2 Hybrid recommender systems in education

The recommender systems have been realized in the education field, specifically in e-learning and academic guidance. The users such as graduates, students, instructors' trainers, or educational counselors use RS to get appropriate recommendations such as courses, training, universities, and university majors. The authors (Farzan and Brusilovsky, 2006) worked on developing a RS based on an adaptive community to recommend appropriate courses to active learners. Thus, they analyzed learners' career goals by implementing a social navigation technique. Besides, (Bendakir and Aïmeur, 2006) suggested a RS approach used to analyze closeness between university course programs and students' profiles. This approach implemented the association mining rules technique for recommending appropriate tasks to learners. As well, Protus (Klašnja-Milićević et al., 2011) a programming tutoring RS can be adapted to the learners' knowledge levels and interests. This system can identify all patterns of the learning models and learners' behaviors based on the learners' learning styles and mining. Protus system generates te clusters by analyzing the different learning styles. Then, studies the learners' behaviors and the interests by mining the frequent sequences using the AprioriAll procedure. Protus RS generates the learning content personalized recommendations based on the ratings of the frequent sequences.

Several hybridization models have been implemented with the combination of the *CF* and *KB* recommender systems to improve the accuracy of recommendations. The possibility of combining *CF* and *KB* techniques is introduced in (Burke, 1999). This hybridization approach has the ability to overcome CF limitations such as the problem of new users or items.

As well as, many researches proved the efficiency of the KB recommender system in the e-learning domain. As an example, (Chavarriaga et al., 2014) proposed a CF and KB technique to suggest learning resources or activities. This approach helps learners to reach high competency ranking by using an online course platform. Experiences or cases are saved and rated by past learners and then integrated into the hybrid RS process. The proposed RS studies learner's capability levels and compares them to past learners' performance in order to find similarities between users' profiles. Then the system offers successful learning advice to active learners.

Moreover, (J. K. Tarus et al., 2017) proposed a KB hybrid RS supported by ontology and *sequential pattern mining* (SPM). This hybrid system provides recommendations of e-learning resources to learners. In this system, the ontology is integrated to represent the domain knowledge about the learner and learning resources. The role of SPM algorithm is to determine the learners' sequential learning patterns. This approach includes four phases: firstly, creating the ontology to represent the learner and learning resources knowledge. Secondly, calculating ratings similarity based on ontology and providing recommendations for the active learner. Thirdly, generating the top N learning products by the CF engine. Finally, implementing the SPM to the top N learning products to provide the final recommendations.

As well, (Ibrahim, n.d.) designed and implemented a hybrid RS named OPCR that incorporates the *CB* filtering and *CF* filtering supported by ontology to overcome the user Cold-start problem. This study presented a novel ontology-based hybrid RS approach that recommends personalized courses that match student's personal needs. This system incorporates all information about courses and helping students to choose courses towards their career goals. The ontology similarity with rating values was used in the *CF* to enhance the ability of the KNN algorithm to find the top nearest neighbor of the active user.

Besides, (Hsu, 2008) presented an online-personalized English learning RS. This system is capable of suggesting students with reading lessons that fit their interests. This hybrid RS incorporates the CB, CF and data mining techniques in order to studies students' reading data and computes recommender scores. In conclusion, this RS has demonstrated to be beneficial in increasing the learners' motivation and their interest in reading.

Additionally, (Rodríguez et al., 2015) presented a student-centered *Learning Object* (LO) RS based on a hybridization approach that incorporates the *CB, CF*, and *KB* techniques. The LOs that are adapted to the learner model/profile are retrieved from the LO databases by implementing the saved descriptive metadata of the objects. This system proved the effectiveness of implementing this type of hybridization in the e-learning domain.

Furthermore, (J. Tarus et al., 2017) proposed an approach that combine the CF and KB supported by ontology to suggest personalized learning materials to online learners. In this system, the ontology is used to represent the learner characteristics while CF predicts ratings and provide recommendations. The ontology is used at the primary phases in the absence of ratings to reduce the cold-start issue. The experiments evaluations showed that the proposed hybrid system outperforms the CF technique in terms of recommendation accuracy and personalization.

This section presents many hybrid RS approaches that were applied in the education domain. These approaches describe many hybridization forms such as the KB with CF techniques, CB with CF techniques, and CB with KB and CF techniques. Besides, our proposed KB hybrid RS (COHRS) differs from these approaches in many aspects. Since, COHRS incorporates four core techniques namely the KB, user-based CF, CBR, and ontology. In addition, all the mentioned approaches were applied to recommend to learners only learning resources. However, COHRS recommend universities, university's majors, and career domains. Moreover, COHRS recommendations are provided to high school students and not to e-learners or university students. Finally, COHRS is supported by the CBR and ontology that allow the system to efficiently retrieve the most similar prior graduates' cases from the case base. In the next section, the related works of the Ontology-based recommender systems that are applied in the education domain are presented.

2.3.2.3 Ontology-based recommender systems in education

As well, ontology-based RS approaches are very popular in the e-learning domain due to their capability to cluster learner models based on their educational background, learning style, study trajectory, and knowledge level. Numerous ontology-based RS have been proposed and implemented with the association of many different recommendation techniques.

In this paper (Shishehchi et al., 2012), authors built an ontology-based RS to recommend suitable materials to learners. The ontology used in this system integrated learners and learning materials knowledge. Besides, the authors developed the "*pedagogy pattern*" due to the importance of pedagogy in learning excellence. Their RS performs based on this pedagogy. In addition, in this paper (Qiyan Han et al., 2010) the authors proposed an ontology-based learning material RS. This recommender is composed of three main mechanisms: semantic rules, ontology design, and concept lattices clustering. Also, the authors in this paper (Capuano et al., 2014), built an adaptive e-learning RS called "*IntelligentWebTeacher* (IWT)". This system combined CF and KB techniques supported by the ontology. Likewise, this paper (Pukkhem, 2014) presented "*LORecommendNet*", which integrated the ontology in their RS to model the learner profiles, material knowledge, and semantic mapping rules. Similarly, this paper (Zhuhadar and Nasraoui, 2010) presented a hybrid RS based on multi-ontology to suggest e-learning content. Moreover, the authors in this paper (Shen and Shen, 2005) presented an ontology-based RS supported with sequencing rules to recommend smart recommendations. On the other hand, this paper (Zhang,

2013) proposed an ontology-based CF technique that uses nearest neighbors to find similar users by computing the similarities using users' ratings. On the other hand, the paper (Huang et al., 2011) presented an ontology-based personalized RS for studying experiences, studying trajectories, studying contents. In this system, the ontology is used to represent the courses, LOs, and learners' knowledge. The results showed that this system improved the learning performance of the learners. In addition, this paper (Blanco-Fernández et al., 2011) presented a hybrid system that mixed the CB with spreading activation and semantic association techniques supported by the ontology. Ontology was integrated into this system to discover knowledge about learners' preferences. In addition, this paper (Mota et al., 2014) proposed an ontology-based RS approach that helps instructors in creating teaching and learning activities. This paper (Rani et al., 2014) proposed a hybrid mechanism, which combines ontology and fuzzy-logic techniques for answering the semantic questions. Moreover, the authors in this study (Cobos et al., 2013) proposed a hybrid RS supported by ontology called "*Recommendation System of Pedagogical Patterns* (RSPP)". This hybridization mixed CF and CB techniques in order to allow instructors to identify significant teaching strategies and apply them in a specific lesson or class. In the next section, the related works of the CBR-based recommender systems that are applied in the education domain and several other domains are presented.

2.3.2.4 CBR-based recommender systems

CBR systems are models for reasoning then learning through experience appropriate for recommender engines. Thus, CBR system is a useful tool to support recommender systems in their recommendation process. In most CBR-based recommender systems, the case-base retains the items to be recommended and the set of recommended items is retrieved from the case-base by matching items similar to that defined by the active user (Burke, 2000).

Many studies have been conducted in the CBR-based recommender systems area. For example, *Entree* (Burke, 2000) is a restaurant RS that generates suggestions by retrieving restaurants in different city similar to the user's tastes or goals. The active user interacts with this RS by stating a visited restaurant in a city and requesting a similar one with the same attributes. Then the system chooses from the case-base the collection of all restaurants that match the active user's constraints. The active user may accept the recommendations and the RS process finishes or he/she criticize the suggestions. The recommendations may not match the user's requirement due, for example, to the high price of products and services of the recommended restaurants. This restarts the recommendation process by considering the criticized features as a significant user aim.

Similarly, (Fesenmaier et al., 2003) is a CBR travel RS that assists the user in organizing a vacation in a decided location. This RS incorporate three recommendation techniques namely the Single Item Recommendation (SIR), Seeking for Inspiration (SI), and travel completion (TC*).* In SIR, the user can cooperates with the RS by querying suggestions about an item category. SIR system asks the user some general preferences and some specific item preferences that are used for querying

the item directory. In TC, the RS suggests extra travel services/items in order to finish the existing user's travel plan. In this technique, the process begins with the new situation as a source case. The source is used by the system in order to retrieve the most similar cases. In the third technique SI (Ricci et al., 2005), the user is prompted with all the travel suggestions to choose from. SI acts as a loop that begins with a source situation and ends when the active user chooses one of the suggested cases.

Moreover, the Order-Based Retrieval (OBR) approach incorporates many types of criteria (Bridge and Ferguson, 2002). The authors developed a RS based on OBR that assists the user to rent a place in London. In this system, the cases are modeled based on a content model. In addition, the attributes were categorized as ordered or unordered values. The process begins when the active user enters his/her preferences and tastes. Then, in the retrieval process, the user input are converted into order relationships on the attributes. Finally, the order relationships are merged in a preorder to generate a lattice of items.

Also, Expertclerk is a CBR-based recommendation system that implements virtual salesclerk systems for e-commerce sites (Mougouie et al., 2003). In this system, the question section technique is implemented: the recommendation process begins by asking the active user some questions. These questions are structured in a decision tree as nodes. Here, the CBR system concatenates the user's answer nodes and creates a SQL-retrieval condition. Then, the CBR system ranks the items in the revise process, suggests items, and clarifies their features. This system switches may allow the user to refine his/her query. Once the refining process ends, the CBR system implements the current query to the case-base and finds new cases. Finally, the retrieved cases are ranked and posted to the user and the process continues awaiting the user to selects a satisfying item.

As well, (Cunningham et al., 2001) presented the WebSell system that applied the CBR system specifically the retrieval and adaptation methods for item recommendation systems. In this system, two item selection lists represent the user profile, which is integrated into a CF recommendation process. These two lists cover both interesting and uninteresting items that are used as a case-base to recommend new items to the active user. The user profile can save information such as preferences and personal information.

Furthermore, (Burke et al., 1997) implemented a CBR-based RS that retrieves products that fit some constraints and rank the returns according to certain criteria. This RS generates recommendations by retrieving restaurants in a city similar to restaurants the active user likes. Finally, (Göker and Thompson, 2000) presented an advanced method to the CBR retrieval process based on diversity that manages the replication of the same suggestion to the active user. Therefore, this system avoids suggesting in a short period the same restaurant to the active user.

CBR's major tasks involve the utilization of specific and general domain knowledge, problem identification, problem-solving, experiences' learning, merging different reasoning techniques.

The core originality of our hybrid RS compared to all of these previous systems is the integration of CF and KB filtering technique supported by ontology technology and CBR system.

- **CBR-based recommender systems in education**

CBR-based recommender systems are solutions for the ever-growing e-learning resources. Unlike other recommender systems, a CBR recommender system does not need to save an enormous volume of data about products rating or specific users. The CBR is a specific information retrieval technique extensively used in nearest neighbor recommender systems. Several CBR-based RS approaches in the education domain have been proposed by implementing various recommendations techniques.

For example, (J. Sandvig and R. Burke, 2005) proposed a system named "Academic Advisor Course Recommendation Engine" (AACORN) which implements the CBR technique based on the knowledge of past cases. This system retrieves solutions to solve future cases. The proposed system integrated knowledge such as past students' experience and the courses' histories to guide learners in choosing appropriate courses.

Also, this paper (Gil et al., 2012) proposed a novel technique to compute educational content discovered by a CBR system. This system is named AIREH (Architecture for Intelligent Recovery of Educational content in Heterogeneous Environments) that can retrieve and incorporate varied educational content. This recommendation approach and the applied technologies in this research were implemented on educational content. Additionally, the used techniques are examples of the possible personalized labeled educational content acquired from diverse environments. The architecture of the AIREH system offers many viewpoints to evaluate the retrieval of educational content from a real environment.

Besides, this paper (Bousbahi and Chorfi, 2015) proposed a CBR-based RS that recommends adequate MOOCs in reply to a particular request of the learner. This system use a special retrieval information method in order to propose to active learner the most suitable MOOCs from different resources based on his/her profile, requirements and knowledge. As well, this work (Duque Méndez N.D. et al., 2018) proposed an assistant recommender system aimed to guide learners in choosing educational material from the database. This RS is based on the CBR artificial intelligence approach. Using old cases of learners with similar preferences and characteristics, this improved the recommendations for each active learner. Moreover, the authors in this paper (Bousbahi and Chorfi, 2015) proposed a CBR RS to address the learners' difficulty in finding courses that best fit their personal interests. The authors implemented CBR with an information retrieval approach to suggest required MOOCs that fit learners' queries. This system computation is based on learners' needs, knowledge, and profile model.

Furthermore, this research (Gomez-Albarran and Jimenez-Diaz, 2009) proposed a CBR approach for the personalized recommendations and learners' authoring tasks in online repositories of

Learning Objects (LOs). This approach combined CB filtering with CF mechanisms. Learners' authoring tasks included the integration of ratings of the existing and new LOs. This RS will be applied to more than 200 programming examples. The next section presents the CBR and ontology-based recommender systems related works.

2.3.2.5 CBR and ontology-based recommender systems

Few studies have been conducted in the CBR and ontology based recommender systems area. For instance, this study (Kowalski et al., 2013) presents as an example a RS for the acquisition and reuse of knowledge about international transport projects that is based on relevant ontologies integrated in the CBR cycle. The aim of this study is to support the intelligent semantic content-addressed reuse of knowledge about such projects.

Besides, this paper (Andreasik, 2017) presents the architecture of a system recommending preventive/corrective procedures in the occupational health and safety management system. The system includes four modules: Module A — an ontology of the workplace OHS profile, Module B — an ontology of preventive/corrective procedure indexation, Module C — a recording system of the monitoring process of non-compliance with the requirements of OHS, Module D — a recommending engine consistent with the CBR concept. This approach integrates the monitoring system of the analysis process of non-compliance with the requirements of OHS at the workplace with the CBR process. The next section presents the comparative analysis for COHRS and other hybrid recommender systems.

2.4 Synthesis

In this section, the selection of the recommender systems for the comparative analysis is based on many criteria such as comparing only hybrid systems that incorporate at least two recommendation techniques like the KB, CF or CB. In addition, these hybrid systems should be used specifically in the education domain and target only learners, not teachers. Moreover, these hybrid systems should be supported by ontology, CBR systems or other technologies. Furthermore, these hybrid systems should integrate data such as ratings, demographics, or knowledge related to the education domain. By applying the mentioned criteria, the selection process leads to select the six hybrid recommender systems that are shown in the following comparative analysis table.

The following three tables present the comparative analysis for *COHRS* and other hybrid recommender systems. The following table illustrates the hybridization strategy, key feature, targeted users and targeted domain of the compared hybrid recommender systems.

Hybrid RS Name	***Hybridization Strategy***	***Used for***	***Targeted users***	***Targeted domain***
COHRS	KB + User-based CF	Guiding high school student toward higher education	High school students	Schools and Higher education

		choices such as university and university majors.		
Chavarriaga et al. Hybrid RS (Chavarriaga et al., 2014)	KB + Item-based CF	Recommending activities and resources that help students in achieving competence levels throughout an online course.	Online learners	E-learning
Mohammad I. (Ibrahim, 2019) OPCR	CB + Item-based CF	Recommending personalized courses that match student's personal needs.	University students	Higher education
Jhon K. Tarus et al. Hybrid RS (J. K. Tarus et al., 2017)	KB + Item-based CF	Generating recommendations of e-learning resources to learners.	Online learners	E-learning
Hsu Mei-Hua Hybrid RS (Hsu, 2008)	CB + Item-based CF	Recommending students with English reading lessons that fit their interests.	Online learners	E-learning
Rodriguez et al. Hybrid RS (Rodríguez et al., 2015)	CB + CF + KB	Providing learners with appropriate recommendations adapted to their preferences and bringing LOs closer than expected.	Online learners	E-learning
Tarus et al. Hybrid RS (J. Tarus et al., 2017)	CF + KB	Suggesting personalized learning materials to online learners.	Online learners	E-learning

Table 2-6: The hybridization strategy, key feature, targeted users and targeted domain of the hybrid recommender systems.

The following table describes the support techniques used by each hybrid RS such as the CBR and ontology. In addition, it shows two significant features that can be used by a hybrid RS such as "*can learn?"* and "*can treat high dimensional datasets". Can learn* means that the system is smart enough to learn from past transactions and experiences. Whereas, "*can treat high dimensional datasets"* means that the system can compute datasets that encompass a high number of attributes.

Hybrid RS Name	***Supported By CBR***	***Supported by Ontology***	***Can Learn?***	***Can treat high dimensional datasets***
COHRS	X	X	X	X
Chavarriaga et al. Hybrid RS (Chavarriaga et al., 2014)			X	
Mohammad I. (Ibrahim, 2019) OPCR		X	X	
Jhon K. Tarus et al. Hybrid RS (J. K. Tarus et al., 2017)		X	X	
Hsu Mei-Hua Hybrid RS (Hsu, 2008)			X	
Rodriguez et al. Hybrid RS (Rodríguez et al., 2015)			X	

Tarus et al. Hybrid RS (J. Tarus et al., 2017)		X	X	

Table 2-7: The core techniques implemented by each hybrid RS

The following table shows the required data by each hybrid RS to generate recommendations. The difference between CORS and other hybrid RS is that CORS can integrate and compute heterogeneous data types such as ratings, knowledge and demographic data, etc. Whereas, other hybrid RS can treat other types of data such as content data.

Hybrid RS Name	***Ratings***	***Domain knowledge***	***Demographic data***	***Content data***
COHRS	X	X	X	
Chavarriaga et al. Hybrid RS (Chavarriaga et al., 2014)	X	X		
Mohammad I. (Ibrahim, 2019) OPCR	X	X		X
Jhon K. Tarus et al. Hybrid RS (J. K. Tarus et al., 2017)	X	X		
Hsu Mei-Hua Hybrid RS (Hsu, 2008)	X			X
Rodriguez et al. Hybrid RS (Rodríguez et al., 2015)	X	X		X
Tarus et al. Hybrid RS (J. Tarus et al., 2017)	X	X		

Table 2-8: The required data by each hybrid RS

2.5 Conclusion

This chapter has discussed the study fields of this research, which encompass basic types of recommender systems such as CF, CB, DF, KB, and hybrid systems. Also, the required data for each RS technique and recommender systems limitations and problems are presented. Additionally, this chapter presents several related works that describe previous researches related to ontology-based recommender systems in education, CBR-based recommender systems in education, hybrid recommender systems techniques, FCA clustering technique, etc.

Our general study of the RS techniques can be summed up by the advantages and drawbacks of the hybridization approaches shown in the following table.

Hybridization model	***Advantages***	***Drawbacks***
KB + CF Memory-based	• Reduced the cold-start problem • Reduced the sparsity issue • The accuracy of the recommendations of this hybridization outperforms the memory-based CF predictions. • Fast replying when user's preferences are modified.	• It is not scalable for large datasets. • It needs knowledge engineering.

KB + CF Model-based	• The accuracy of the recommendations of this hybridization outperforms the model-based CF predictions. • It has a scalability feature. • Fast replying when user's preferences are modified.	• It needs knowledge engineering. • The hybridization of the memory-based CF with the KB provided better results than this system.
DF + CF Memory-based	• Reduced the cold-start problem. • The accuracy of the recommendations of this hybridization outperforms the memory-based CF predictions.	• It is hard to acquire demographic data. • It is not scalable for large datasets.
DF + CF Model-based	• The accuracy of the recommendations of this hybridization outperforms the model-based CF predictions. • It has a scalability feature.	• Find difficulty in obtaining demographic data • The hybridization of the memory-based CF with the DF provided better results than this system.

Table 2-9: Comparison of the hybridization techniques

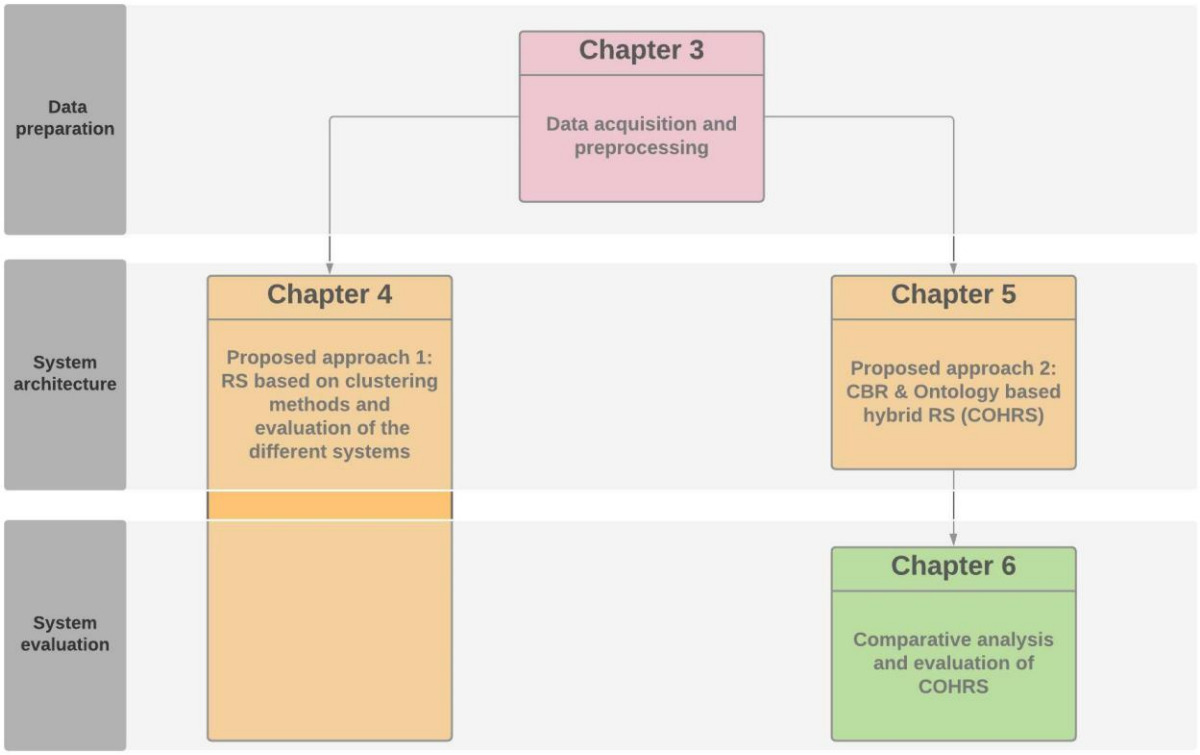
In summary, our approach differs from previously mentioned approaches in the sense that it combines user-based CF with KB techniques supported by CBR and ontology. This approach is named *CBR and ontology based hybrid recommender system (COHRS)*. This hybridization aims to improve the recommendations and precision of the system. In this hybrid system, we used the ontology engineering to model the knowledge acquired from different resources such students' demographic data, interests, schools, universities/colleges, university majors, and career domain. The CBR and ontology are integrated into COHRS to solve the traditional recommender system limitations. COHRS is specialized in the field of guiding high school students toward higher education paths. No study has been conducted to describe the higher education domain with a hybridization strategy that combines the KB, CF, ontology, and CBR techniques. Finally, the architecture of our hybrid RS approach is described in detail in chapter 5. The next chapter presents the data acquisition and preprocessing phases of our study.

PART III

CONTRIBUTIONS: PROPOSED APPROACHES FOR THE RECOMMENDATION OF A UNIVERSTIY PATH

In this part, chapters 3, 4, 5, and 6 are presented. These chapters present our contributions and proposed approaches for analyzing individual educational trajectories and recommending via a hybrid RS personalized recommendations such as a university, university major and career domain.

The following figure illustrates the organization of these chapters that are ordered in three main sections namely (*Data Preparation, System Architecture, and System Evaluation*). Section *Data preparation* includes chapter 3 that presents the data acquisition and data preprocessing techniques. Section *System Architecture* includes chapters 4 and 5. Chapter 4 presents our first proposed approach, whereas chapter 5 presents our second proposed approach. In chapter 4, we proposed to implement different clustering techniques in order to cluster the university graduates' trajectories and recommend university paths based on the results of the clusters. In chapter 5, we proposed a hybrid system approach named COHRS that is based on CBR and ontology, which recommend personalized recommendations to high school students. Finally, section *System Evaluation* includes chapter 4 where we evaluated the proposed clustering techniques and chapter 6 that presents comparative analysis and evaluations for COHRS and other RS approaches.



CHAPTER 3: Data acquisition and preprocessing

This chapter presents the preparation and dissemination of our online survey, data acquisition, and data pre-processing techniques.

3.1 Introduction

Nowadays, data mining, data analysis, recommender systems, machine learning, artificial intelligence, etc. are important technologies. However, they do not function without the essential fuel, which is the data.

As well, our study focuses on analyzing and clustering university graduates trajectories and finding solutions to assist high school students to take appropriate decisions toward higher education choices. However, this study requires special types of data to be used in the analysis phase. Unfortunately, the required data is not available anywhere, since it is related to university graduates educational trajectories. In addition, it is very difficult to acquire it online and users are reluctant to disclose it.

Data can be acquired in two ways: implicitly or explicitly. Explicit data are acquired from user ratings; for example, after listening to a song or watching a movie, and the implicit data are acquired form purchase history, search engine searches, or users/items' knowledge.

In our case, we worked on gathering the required data through the explicit method. Therefore, we disseminated an online survey that includes 55 questions. The survey purpose is to reach university graduates and collect information about their educational trajectories, interests, current career occupation, etc. The survey was created in bilingual form (English and French). The dissemination process of the survey covered the Lebanese university graduates.

In the next sections, the preparation and dissemination process of the online survey, data acquisition, and data pre-processing are presented.

3.2 Preparation and dissemination process of an online survey

Our online survey has targeted the Lebanese university graduates that pursued different university major. This survey was published online in a period of 6 months. The survey involved more than 50 questions of heterogeneous data types such as nominal, ordinal, numerical and open-ended.

Ordinal data take their values in an ordered finite set. For example, a survey may ask the user to provide feedback on the service he/she received in a restaurant. The quality of service is ranked as (1) Not at all Satisfied, (2) Partly Satisfied, (3) Satisfied, (4) More than Satisfied and (5) Very Satisfied. The larger the set of values, the more informative the data.

Nominal data names somewhat without assigning it to an order in relation to other numbered items of data. For example, "acting", "camping" or "cycling" classification for each user's hobbies.

Numerical attributes with continuous values that are represented by numbers and have most of the characteristics of numbers.

Open-ended questions are questions that ask an applicant to answer in their natural language. They require a longer response. Thus, *open-ended* questions provide more information than a simple yes or no answer.

Our survey collected a real-world dataset that includes about 1000 university graduate applications and approximately 20,000 high school course ratings. This real-world dataset has varied data such as demographic data, interests, education and career knowledge, and ratings.

A collection of question types was used in this survey such as multiple-choice, Likert scale, and open-ended questions. For example, the answers for "How would you rate your high school grades on the following (Biology, Chemistry, Physics, and Mathematics…)?" are *Very Good, Good, and Poor/Not concerned*. Similarly, answer options for "If you already changed your university major, why did you change it?" are *badly advised, Lack of understanding, you were uninterested in courses,* and *you had new interests*...The following figure shows a screenshot of a Likert scale question taken from our online survey.

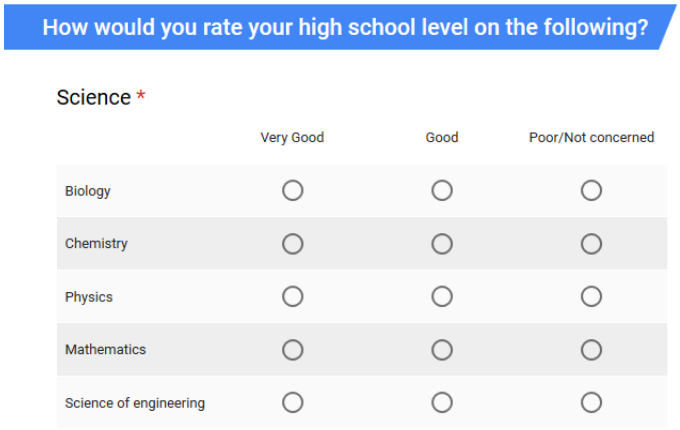


Figure 3.1: Likert scale sample

As mentioned before, this survey has targeted university graduates in order to collect data about the graduates' educational trajectory. The survey was organized into six main sections namely the survey description, graduate personal information, graduate high school or vocational school information, graduate first attended university information, graduate interests, and career information, and graduate current university major information.

Since our hybrid RS recommends universities, university majors, and career fields, the following four criteria were applied to create our survey sections and questions.

First criteria: the survey should include graduates' family information, demographics, and personal data such as gender, hobby, language, etc., in order to recommend personalized recommendations. Therefore, the survey includes the "Graduate personal information" section.

Second criteria: the survey should include graduates' high school or vocational school data such as graduates' school courses interests, school sector, school education system, etc., in order to recommend to high school students recommendations based on their high school information. Therefore, the survey includes the "Graduate high school or vocational school information" section.

Third criteria: the survey should include graduates' university information such as teaching effectiveness, their university major, university name, etc., in order to recommend to high school students recommendations related to university paths. Therefore, the survey includes the "Graduate first and currently attended university information" sections.

Fourth criteria: the survey should include graduates' career information such as their current occupation, career interests, etc., in order to recommend to high school students career choices related to their career interests. Therefore, the survey includes the "Graduate interests and career information" section.

All these criteria have covered the graduates' trajectories starting from studying at high school, then studying at the university, then entering the career market. Integrating university graduates' trajectories data in our hybrid system recommendation process helped us to recommend to high school students universities paths, and career choices.

The following are some samples of questions copied from the survey sections:

- **Graduate personal information section**
- What is your Gender? (Male or Female)
- Select the work of your father
- **Graduate high school or vocational school information section**
- What high school did you attend? (High School or Private School)
- What high school subject did you like best?
- **Graduate first attended university information section**
- What was your university major?
- How effective was the teaching within your major at the university? (Very Effective, Somewhat Effective or Not So Effective)
- **Graduate interests and career information section**
- What kind of job/career interests you?
- Is your current job related to your university major?
- **Graduate current university major information section**
- What degree are you currently pursuing? (Bachelor Degree, Master Degree, Doctoral Degree or Other)
- How many times (if any) did you change your major at university (current or before)?

In this section, simplified examples of the survey questions are presented. More details about the survey's sections and questions are presented in Appendix B. In the following section, details about the data-preprocessing phase are presented.

3.3 Data preprocessing

Inappropriate and redundant information or unreliable and noisy data exist in all dataset. Thus, analyzing data that has not been carefully refined can generate inaccurate results. Here comes the role of data preprocessing that involves many important steps and techniques. Therefore, data preprocessing is an essential phase in the data mining process and machine learning projects. In our work, we applied the following data preprocessing techniques in order to clean and refine the acquired data from the online survey.

- *Data Quality Evaluation* – data must be checked for missing, inconsistent and duplicate values.
- *Dataset Dimensionality Reduction* – significant real-world datasets have a great number of attributes (features). Therefore, the *dimensionality reduction* technique's purpose is to reduce the number of features in order to make the processing of the data more tractable. Reducing the dimensionality of a dataset is done by defining new features which are an arrangement of the original features.
- *Attribute Sampling* – sampling is picking a subset of the dataset that we are studying. Analyzing the whole dataset can be too expensive considering the time and memory constraints. Implementing a sampling algorithm can aid in reducing the size of the dataset to a level where the analyst can use a better machine learning algorithm.

In order to initialize the data-preprocessing phase, we extracted the required data as a CSV file from the online survey. The following figure illustrates the process of filling, extracting, cleaning, refining and preparing the desired dataset using many data preprocessing methods and tools such as Weka, WordNet, and Levenshtein distance.

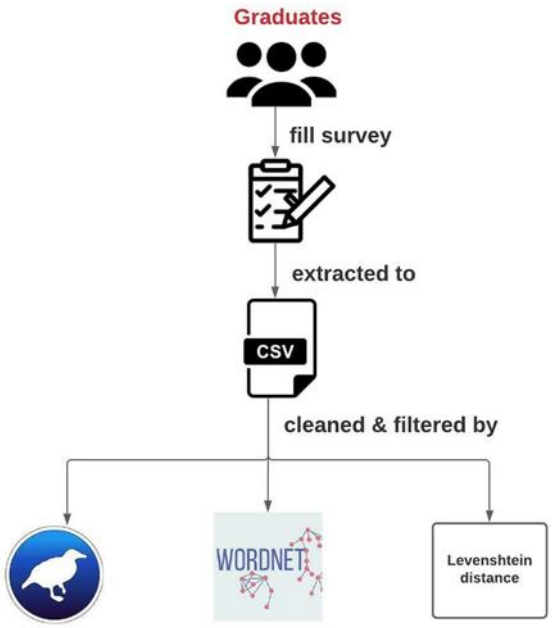


Figure 3.2: Data pre-processing

As mentioned before, our survey contains 55 questions which led us to a large number of features. As the proposed approaches in chapters 4 and 5, especially the implemented clustering techniques need a limit number of features to process, we were obligated to reduce the dimensionality of our dataset. Therefore, we used Weka *InfoGainAttributeEval*[2] technique to perform feature selection by calculating the information gain for each feature for the output variable. The entry values range from 0 (means no information) to 1 (means maximum information). The features that give more information will get a higher information gain value and can be chosen, whereas those that do not show much information will get a lower score and can be ignored in the analysis process.

Additionally, records of duplicate data should be deleted from the dataset before the analysis phase starts. Therefore, the Python *drop_duplicates*[3] function was applied to drop duplicate records from our survey data. Moreover, the multi-answer questions were split, using "," as a delimiter as shown in Table 3-1. Thus, we used the python *Series.str.contains* and *Series.string.split* functions to find specific terms and split each row in the series based a delimited. The following figure shows an example of multi-answer question:

[2] https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html
[3] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html

	What are your outside interests (hobbies)?
Graduate 1	Dancing, fashion, reading
Graduate 2	Swimming
Graduate 3	Football
Graduate 4	Reading
Graduate 5	Camping, swimming
Graduate 6	Sports
Graduate 7	Reading
Graduate 8	Photography
Graduate 9	Playing music, singing, reading
Graduate 10	reading hiking
Graduate 11	Sports, reading
Graduate 12	football
Graduate 13	Cooking, camping, reading
Graduate 14	Tennis

Table 3-1: multi-answer question example

Besides, in the next section we present the *WordNet* lexical database to get the synonym of some terms and strings that describe the same object. Whereas, in section 3.3.2 the Levenshtein distance application is presented to correct the misspelled terms and strings.

3.3.1 Searching semantic relations with WordNet

WordNet (Fellbaum, Christiane, 2005) is a huge lexical database of many languages lexical. It is formed of sets of synonyms or synsets, which are groups of Nouns, Adjectives, Verbs, or Adverbs. The synonyms are linked based on lexical relationships, such as hyponym, hypernym, antonym, etc. This lexical database is available online for free download and usage. *WordNet's* structure (Jurafsky and Martin, 2009) enables this tool to deal with numerous tasks in *NLP* and computational linguistics such as information and document retrieval, improve search engine returns, automated document and text classification, Word sense disambiguation, machine translation, online lexical dictionary, etc.

WordNet is usable from the *R* language to compute linguistic and text mining processing. As explained before, our survey contains open-ended questions. Thus, the graduates answered these questions in natural language and they expressed the same information differently. For instance, the term bike could be expressed as bike, bicycle, motorcycle, wheel, and cycle. In order to regroup our data, we used the *WordNet* database to find the synonym of the terms and strings that were entered by the graduates in their natural language.

Getting the synonyms is required in the data prepressing phase in order to find the meaning of the terms and strings that describe the same object and then unify it in one common term. For example, the “IT manager” and “Information Technology manager” represent the same career domain. However, the clustering techniques will consider the “IT Manager” and “Information Technology Manager” as two different strings. Therefore, unifying the terms or strings into one common term

can help the clustering technique to consider it as a same object and cluster it in the same group. A second example, the terms teacher and instructor share the same meaning but they are considered as two different terms in the clustering process. Nevertheless, when the synonym of the two terms is retrieved using *WordNet* and then unify it in one common term, the clustering technique will cluster it in the same cluster.

Table 3-2 shows the examples and results of our R system that uses the *WordNet* lexical database. In this figure, the "*Source*" column represents the terms and strings that were entered by the graduates and columns *WordNet* result 1, *WordNet* result 2, *WordNet* result 3, and *WordNet* result 4 are the related synonyms found in *WordNet*.

Source	**WordNet result 1**	**WordNet result 2**	**WordNet result 3**	**WordNet result 4**
Translator	Interpreter	transcriber	Translating program	Translator
Editor	Editor	editor in chief	editor program	
Physical Therapist	physical therapist	physiotherapist		
Police Office	Officer	police office	policeman	
Teacher	Instructor	Teacher		
IT engineer	not found			
Social media manager	not found			
Music teacher	music teacher			
Waiter waitress	not found			
IT manager	not found			
Internal security forces	not found			
Market research analysts	not found			
Accountant	Accountant	comptroller	controller	
Data scientist	not found			
Management	Direction	management		
Accountant	Accountant	comptroller	controller	
Banker	Banker			
Civil engineer	civil engineer			
Sales manager	not found			
Operations manager	not found			
Human Resources Manager	not found			
Web developer	not found			
Purchasing Manager	not found			
Assistant prof	not found			

Table 3-2: WordNet synonyms returns

We deduced from our *WordNet* experiments that this lexical tool is more effective in providing synonyms for the given terms. However, it shows a major drawback in retrieving synonyms for the strings that are composed from many terms given in the "*Source"* column. As shown in Table 3-2 the *WordNet* tool could not find synonyms for the strings (*IT Engineer, Social Media Manager, Music Teacher, IT Manager, Internal Security Forces, Market Research Analyst*, etc…). In

addition, *WordNet* misses some potential synonyms because of the spelling errors. The following table shows some misspelled terms and strings at line 3, 4 and 7.

	Source
1	Translator
2	mechanical engineering
3	IT manager.
4	Wen developer
5	Graphic design
6	Teaching
7	Information techno
8	Senior Accountant
9	Media monitoring at ipsos
10	Industrial pharmacist

Table 3-3: Misspelled terms and strings found in our online survey

Therefore, in the next section we present our implementation of the *Levenshtein distance* in order to correct the misspelled terms and strings found in our survey.

3.3.2 Correct misspelled terms and strings with the levenshtein distance

Many misspelled terms and strings were found in our survey entered by the university graduates. Therefore, we implemented the Levenshtein distance in order to compute the match between correct and incorrect terms and strings.

In 1966 (Levenshtein, 1966), the *Levenshtein* string metric was proposed by Vladimir *Levenshtein*. *The Levenshtein distance* measures the dissimilarity between two sequences. The distance between two strings is the least number of single-character alterations needed to transform one string into the other. This metric is applicable in sequence matching and spell checking.

The following table present the *Levenshtein distance* edit actions:

Action	***Description***	***Example***
Insertions	Insert a single character anywhere	Tet > Te**s**t
Deletions	Delete a single character	Tes**e**t > Test
Substitutions	Replace a character by another one	**T**est > **R**est
Transpositions	Change the order of two consecutive characters	Te**st** > Te**ts**

Table 3-4: Levenshtein distance edit actions

Table 3-4 shows edit actions of the Levenshtein distance namely the insertion, deletion, substitution and transposition.

Let's consider two words (s) and (t); when comparing them, (s) represents the source string, and (t) represents the target string. The distance between (s) and (t) is the minimum number of atomic actions (see Table 3-5 below) required to transform (s) into (t). More the distance is high, more the two words are therefore dissimilar. The following two examples explain the distance computation:

- If (s) is "acting" and (t) is "acting", then LD(s, t) = 0, since the two strings are identical (no action needed).
- If (s) is "abtin" and (t) is "acting", then LD(s, t) = 2.

The following table shows the computation of the Levenshtein distance for the target string "Acting" and source string "Abtin".

			Source				
			A	B	T	I	N
		0	$_i$1	$_i$ 2	$_i$ 3	$_i$ 4	$_i$ 5
Target	A	$_j$1	0	0+1=1	1+1=2	2+1=3	3+1=4
	C	$_j$ 2	0+1=1	1	1+1=2	2+1=3	3+1=4
	T	$_j$ 3	1+1=2	1+1=2	1	1+1=2	2+1=3
	I	$_j$ 4	2+1=3	2+1=3	1+1=2	1	1+1=2
	N	$_j$ 5	3+1=4	3+1=4	2+1=3	1+1=2	1
	G	$_j$ 6	4+1=5	4+1=5	3+1=4	2+1=3	1+1=2 Two operations in total

Table 3-5: The Levenshtein distance algorithm

The returned distance (number of changes = 2) of the *Levenshtein* algorithm is shown in the lower right cell corner of the above matrix. Thus, the term "Abtin" could be transformed into "Acting" through 2 changes, namely the substitution of "b" with "c" and insertion of "g".

As we explained before, the graduates' students filled some questions in our survey in natural language and we detected many misspelling errors. To clean our data and avoid loss information we used the *Levenshtein distance* as presented above. This method allows us to clean our dataset based on a reference dataset by computing the similarity between a source column from our dataset and a target column from the reference dataset. This dataset contains the correct terms and strings. It has been extracted from the internet and prepared before the computation process. It contains columns such as hobbies, jobs, and university majors.

The following table shows an exempt of the *Levenshtein distance* results generated by the *adist()* R function. The *Target list* column represents the jobs in the reference dataset and the *Source list* column represents the graduates' jobs that should be cleaned. The purpose of this method is to

transform the misspelled strings and terms given by the *Source list* to correct strings and terms given by the *Target list*.

	Target list	***Source list***	***Distance***
1	Translator	Translator	0
2	Mechanical Engineering	mechanical engineering	0
3	IT Manager	IT manager.	1
4	Web Developer	Wen developer	1
5	Graphic Designer	Graphic design	2
6	Teacher	Teaching	3
7	Information Technology	Information techno	4
8	Accountant	Senior Accountant	7
9	Media Monitoring	Media monitoring at ipsos	9
10	Pharmacist	Industrial pharmacist	11

Table 3-6: Exempt of the adist() method returns

3.4 Conclusion

The aim of this research work is to propose to high school students a university path based on the graduates trajectories. We presented in this chapter the dataset collected from an online survey, disseminated on Lebanese universities graduates. It includes 55 questions describing the educational trajectories of graduates, their profile, their interests, their current career occupation and future career projects. These questions have heterogeneous data types such as nominal, ordinal, numerical and open-ended. The data-preprocessing phase is essential in data mining and analysis. We have explained in this chapter the applied methods to process each type of the above data types.

Therefore, we focused in section 3.3.1 on correcting the misspelled terms and strings that were acquired from our online survey. The misspelled terms and strings were corrected by the implementation of the *Leventhein* distance concept. In addition, the synonyms of many terms were gathered through the *WordNet* lexical database in order to federate the terms that describe the same object.

In our experiments, the *WordNet* performed effectively in providing the synonyms of the given terms that are composed from one word. However, it shows a major drawback in retrieving the synonyms of the strings that are composed from many words. The *WordNet R* program could not find synonyms for strings (IT Engineer, Social Media Manager, Music Teacher, IT Manager, Internal Security Forces, Market Research Analyst, etc…). In addition, the *WordNet R* program misses finding the synonyms of the misspelled terms that were found in our online survey.

Besides, the *Leventhein* distance helped us in the data-preprocessing phase to find match between the correct and incorrect terms and strings. The *Leventhein* distance was used to transform the

misspelled terms and strings into correct version of terms and strings. This data-preprocessing work helped us to prepare a clean and refined dataset ready for the analysis phase.

The following chapter presents the implementation of three clustering techniques namely the *Formal Concept Analysis (FCA), K-modes,* and *Hierarchical clustering*. These techniques were implemented to cluster the graduates' educational and career trajectories.

CHAPTER 4: First proposed approach based on clustering techniques

This chapter presents the implementation of many clustering techniques such as the *Formal Concept analysis*, *Hierarchical* and K-modes.

4.1 Introduction

Chapter 3 presented how we collected our data, cleaned and filtered it from noisy data. The objective of this chapter is to cluster the university graduates' data gathered from our online survey and to identify their education's trajectories. The goal of clustering the graduates' trajectories is to recommend appropriate and personalized recommendations to high school students based on the retrieved clusters.

The collected data includes many categorical features such as graduate interest, preference, hobby, etc. Categorical features are unordered and discrete in contrast to numerical data. Thus, the numerical algorithms for clustering numeric data cannot be applied to cluster categorical data.

In order to address the issues of clustering categorical data, we aimed to find appropriate clustering techniques that are suitable for the type of data in our dataset. Therefore, we selected the *Formal Concept Analysis (FCA), Hierarchical*, and *K-modes* clustering techniques to be applied and experimented on our dataset.

Clustering (Rodriguez et al., 2019) is a technique that groups similar objects such that the objects in the same group are more similar to each other than the objects in the other group. Based on a distance equation, the closeness of two objects in a space can be identified. The group/cluster quality can be described by its diameter, which is the maximum distance among two objects in the group. Clustering is a significant technique in the data mining process and statistical data analysis. Thus, it is implemented in many areas such as image analysis, pattern recognition, information retrieval, machine learning, and computer graphics.

In this chapter, we present many clustering techniques such as *FCA*, *Hierarchical* Clustering, and *K-modes* that deal with categorical data in order to cluster the university graduates data and identify distinct trajectories. In the next section, the *FCA* implementation and analysis are presented.

4.2 Clustering techniques implementation

In this section, the *FCA), Hierarchical Clustering,* and *K-modes* are presented and experimented.

4.2.1 FCA implementation and analysis

This section, presents the FCA data mining technique that analyses and models the university graduates' profiles.

FCA has been implemented in several domains like medicine, mathematics, computer science, psychology, biology, and linguistics. The aim is to analyze the effectiveness of this method in the educational domain and especially on our collected data.

The following two sections present an overview of the FCA study. The implementation process of the *FCA* technique on Lebanese university graduates is presented in section 4.2.1.1 and the conclusion and future study are discussed in section 4.2.1.2.

4.2.1.1 Conceptual clustering process using FCA technique

As declared in the above sections, our proposed system's purpose is to recommend university majors, careers and universities/colleges to high school students relying on their preferences and university graduates' trajectories. In this study, university graduates are students with a minimum of a bachelor's degree or equivalent. Thus, our aim is to analyze university graduates' data such as interests and preferences during their years of study in high school, their university fields of study, and their careers. A second aim is to cluster graduate students' trajectories through the implementation of the *FCA* technique. The following figure shows the data analysis and clustering phases in our FCA approach.

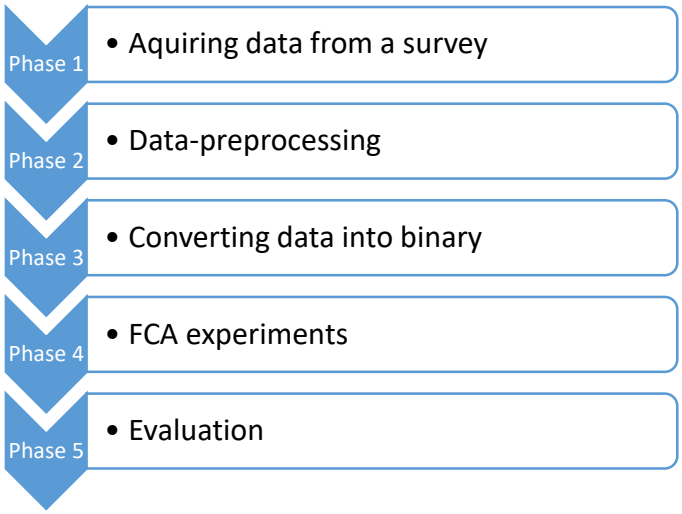


Figure 4.1: FCA approach

Phase 1: Acquiring data from our online survey

This phase is discussed in details in chapter 3. At this point, we disseminated an online survey that contains more than 50 attributes in order to acquire the required data.

Phase 2: Data-preprocessing

This phase is discussed in details in chapter 3. After the data transformation and cleaning process, we refined a dataset of 448 university graduate profiles gathered from different education fields.

Phase 3: Converting data into binary

As discussed in chapter 3, our survey encompasses heterogeneous data types such as demographic data, interests, characteristics, education and career knowledge, and ratings. This survey includes a collection of varied question such as multiple-choice, Likert scale, and open-ended questions. Therefore, in our situation context attributes are non-binary and *FCA* is developed for binary data analysis. This posed the issue of converting data to binary. Thus, we converted all attributes to binary data using special formulas for every attribute in the dataset. The following activity diagram (Figure 4.2) illustrates the algorithm process for converting the school orientation attributes' values into binary:

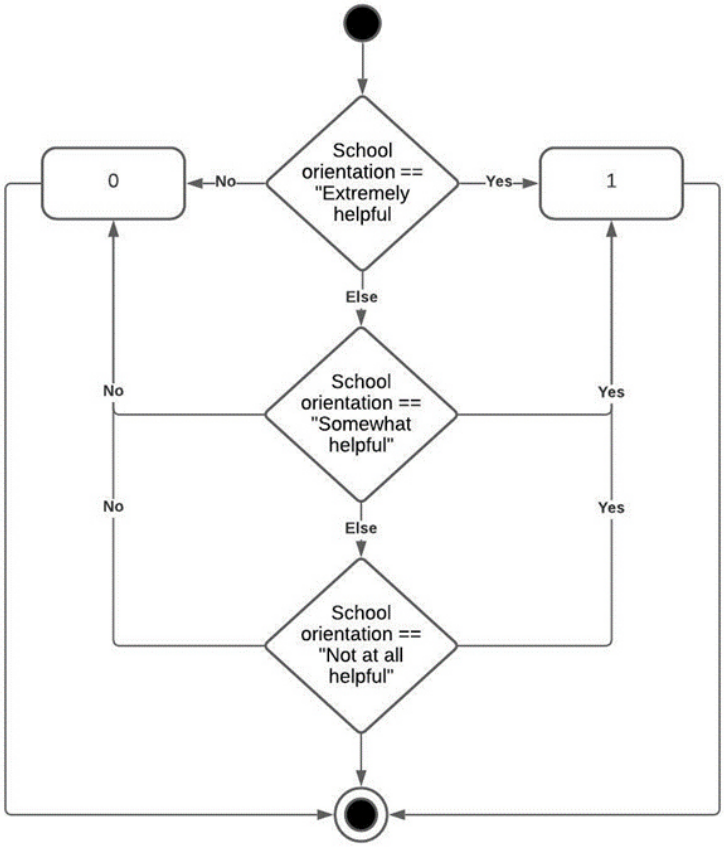


Figure 4.2: Algorithm of converting school orientation value into binary

When using *FCA* for object classification, a scaling problem arises. For instance, the *Ordinal scale* denotes classical real number order. The attributes age can be scaled by this type of scale. In our case, we used three groups for the age. The first group includes individuals with age between 19 and 25 that are usually fresh graduates or still studying at the university. The second group includes individuals with age between 26 and 30 that are usually university graduates or employees in the private or public sector. And, the third group includes individuals with age greater than 30 that are usually expert employees. The following table shows how we scaled the age attribute and converted it into binary via Microsoft Excel If statement with multiple conditions.

What is your age?	Age 19-25	Age 26-30	Age >30
33	0	0	1
30	0	1	0
33	0	0	1
30	0	1	0
30	0	1	0
33	0	0	1
32	0	0	1
37	0	0	1
24	1	0	0
40	0	0	1
31	0	0	1
19	1	0	0
29	0	1	0

Table 4-1: Scaled and converted age values

In addition, *Nominal scale* is suitable for binary representation of nominal (categorical) attributes like hobbies. The hobbies were grouped into four main categories. Each category includes the most similar hobbies or activities. For the context of hobby, the hobbies were scaled by nominal scaling as below:

What are your hobbies?	Arts Hobbies	Sports Hobbies	General Culture Hobbies	Random Hobbies
Cinema	1	0	0	0
marche , lecture	0	1	1	0
Arts	1	0	0	0
Dancing, fashion, reading	1	0	1	0
Music. Sport	1	1	0	0
Swimming	0	1	0	0
Sport, lecture	0	1	1	0
Football	0	1	0	0
Lecture	0	0	1	0
Reading	1	0	1	0
Camping, swimming	0	1	0	1
Piano sport	1	1	0	0
Sports	0	1	0	0

Table 4-2: Scaled and converted hobbies values

More details about the FCA scaling problem are discussed in chapter 2 section 2.5.

Phase 4: *FCA* experiments

This phase shows the implementation of many open-source concept explorer software and algorithms such as (*ConExp* (Serhiy A. Yevtushenko, 2000), *Conexp-ng-0.7.0*[4], *Latviz Loria* (Alam et al., 2016), *Lattice-Miner 2.0*[5] and *In-Close* [6]).

[4] https://github.com/fcatools/conexp-ng/releases.
[5] https://github.com/LarimUQO/lattice-miner.
[6] https://sourceforge.net/projects/inclose/

- ***Our experiments run as follows:***

We imported our binary dataset to the *FCA* Concept Explorer software (*ConExp*), this software displayed all (1) values as X and (0) as a blank cell as shown in Figure 4.3. *ConExp* designated that our dataset includes 448 objects and more than 500 attributes. The 448 objects represent the graduates (G1, G2, G3…) and the 500 attributes represent (Age, Hobbies, High school degree, Gender, etc...). This experiment failed to generate a result and the *ConExp* software could not compute our high dimension dataset. Our dataset features exceeded this software's capability to handle such a number of attributes.

Therefore, we worked on reducing the dimension of the graduates' dataset. To improve the calculation accuracy of data processing, the attributes dimensionality should be reduced. Thus, we implemented the "*InfoGainAttributeEval*" Weka tool to evaluate and rank the dataset features. This tool helped us to select the features that have high significance and produced a dataset of 448 objects and 119 attributes. The new dataset is illustrated in the following figure that is generated by the *ConExp* software.

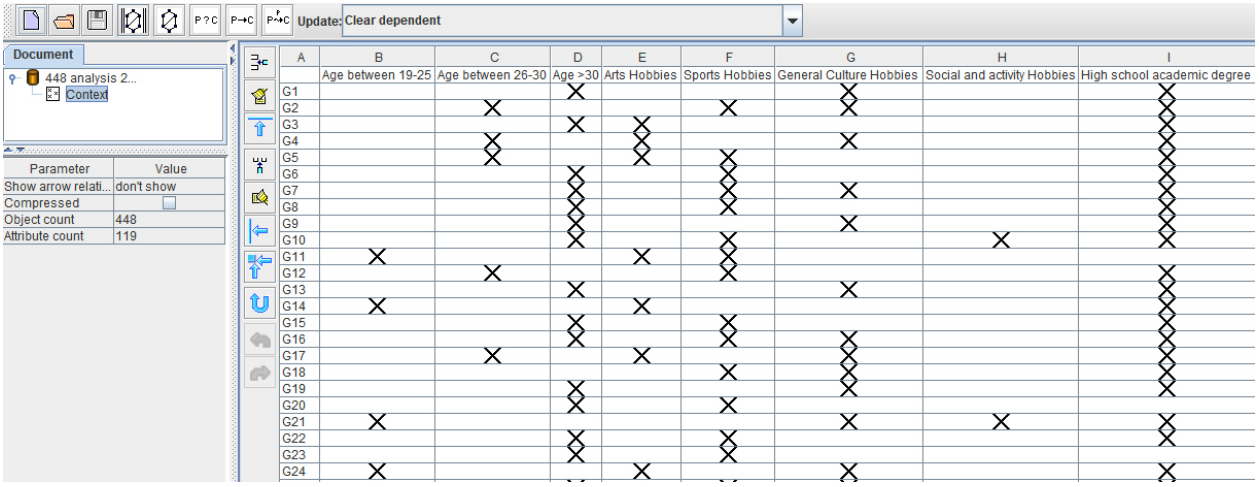


Figure 4.3: Boolean representation in concept explorer software sample 1

Then, we ran many experiments on the new reduced dimension dataset using (*ConExp, Conexp-ng-0.7.0, Latviz Loria and Lattice-Miner 2.0*) software. Once more, the aforementioned software could not produce results due to the high dimension of 119 attributes in our new dataset.

Thus, we changed our course of action to work with the In-Close concept explorer algorithm to cluster university graduates' trajectories. *In-Close* is a Tree maker and fast formal concept miner for *FCA* .cxt files. However, we encountered the same issue where this algorithm could not compute more than 400 objects and 119 attributes per experiment. Once again, we had to reduce the number of objects to run our experiments on *In-Close*. The following figure presents the results of our three experiments realized by the *In-Close* software:

	Input			Output
	Same dataset	Size of intent	Size of extent	Number of clusters
Experiment 1	Yes	27	4	7
Experiment 2	Yes	24	10	9
Experiment 3	No (reduced one)	10	10	4

Table 4-3: The experiments results of In Close

The following three experiments presents the results that were generated by the *In-Close* software:

- ***Experiment 1***

In this experiment, we used a dataset of 400 objects and 119 attributes and we set the following constraints to run the *In-Close* algorithm:

- The minimum size of intent (no. attributes) is 27, meaning the results should group at least 27 attributes per cluster:
- The minimum size of extent (no. objects) is 4, meaning the results should group at least 4 objects per cluster.
- This experiment produced 7 clusters and the following results show one cluster sample:
- *Cluster sample:*
    - Graduate G167, G372, G138 and G133 have in common the following attributes:
    - Their age is greater than 30.
    - They studied business at university
    - They never changed their major at university
    - They are currently employed or have a business
    - Their current job is in the business sector
    - Their current job is related to their university major
    - They rated their performance at school as (Very good or good) in the following courses: Philosophy, Geography, Chemistry, History, Sociology, Sports, Religion, French, Arabic, Physics, Biology, English, Music, Psychology, Economics, Mathematics, Theatre, Technology, Computer Science, and Dance.

- ***Experiment 2***

In this experiment, we used the same dataset, but we set the following constraints to run the algorithm:

- The minimum size of intent (no. attributes) is 24, meaning the results should show at least 24 attributes per cluster.
- The minimum size of extent (no. objects) is 10, meaning the results should show at least 10 objects per cluster.

- This experiment produced 9 clusters and the following results show one cluster sample:
- Cluster sample:
    - Graduates G115, G380, G167, G344, G198, G175, G372, G247, G362, G109 and G226 have in common the following attributes:
    - They are currently employed or have a business
    - Their current job is related to their university major
    - They rated their performance at school as (Very Good or Good) in the following courses: French, Physics, Mathematics, English, Technology and Computer Science, Chemistry, Sports, History, Sociology, Religion, Economics, Geography, Arabic, Philosophy, Biology, Psychology, Foreign language, Drawing, Music, Theatre, Dance.

- ***Experiment 3***

To focus on the university majors and the university graduates' current jobs and situations, we reduced all attributes that rate high school courses and we used more significant attributes. This reduction generated a new dataset of 400 objects and 56 attributes. Then, we set the following constraints to run the algorithm:

- The minimum size of intent (no. attributes) is 10, meaning the results should show at least 10 attributes per cluster.
- The minimum size of extent (no. objects) is 10, meaning the results should show at least 10 objects per cluster.

This experiment produced 4 clusters and the results show more accurate grouping focusing on university graduates' education and career trajectories. The following is one cluster sample:

Cluster example: Graduates G220, G217, G219, G225, G214, G223, G227, G216, G279, G224, G222, G236, G358, G231, G209, and G229 have in common the following attributes:

- They liked scientific courses at school
- They studied Computer Science and Computer communication engineering
- They have a Master's degree
- They are currently employed or have a business
- Their current job is in Computer Science or Computer communication engineering or Information technology
- Their current job is related to their university major
- Their major meets their needs or interests very well
- They are satisfied with their current salary

Any modification on the intent and the extent values will change the number of clusters and the number of attributes and objects in the clusters.

In our case, our dataset contains more than 400 important attributes. However, we had to minimize the number of attributes to be able to run the experiments on the available concept explorer software. Although the dimension was reduced, the experiments showed many issues in computing high numbers of formal contexts. In conclusion, the three experiments of the In-Close did not attain our aim, nevertheless, *experiment 3* generated better accurate grouping focusing on university graduates' education and career trajectories.

Phase 5: Evaluation

This phase shows a comparison, and assessment table for the experimented open-source concept explorer software and algorithms. This table illustrates our experiments' results, software comparisons and assessments:

Software Name	***N. of objects***	***N. of attributes***	***N. of clusters***	***File type***	***Software status***	***Drawbacks***
ConExp	448	119	No result	.cex .cxt .csv	Overloaded No respond	When using this software, users cannot control the required number of clusters in the experiments. Usually, this software arranges large numbers of clusters.
Lattice-Miner 2.0	400	119	No result	.cex	Overloaded No respond	When using this software, users cannot control the required number of clusters in the experiments. Usually, this software arranges large numbers of clusters.
In-Close4	448	119	No result	.cxt	Overloaded No respond	The software does not allow the user to control the clusters of results. Whereas, users need results related to some specific domain in the dataset.
In-Close4	400	119	The results depend on the intent and the extent constraints specified by the user. In our experiments, the software mined (4, 7 and 9) clusters.	.cxt	Ran normally	The software does not allow the user to control the clusters of results. Whereas, users need results related to some specific domain in the dataset.
In-Close	400	56	In this experiment, the software mined (4) clusters.	.cxt	Ran normally	The software does not allow the user to control the clusters of results. Whereas, users need results related to some specific domain in the dataset.

latviz.loria.fr	400	119	No result	.json	Overloaded No respond	When using this software, users cannot control the required number of clusters in the experiments. Usually, this software arranges large numbers of clusters.

Table 4-4: Concept explorer software comparisons and assessments

As a summary result, we deduced that the existing concept explorer systems could analyze and illustrate concept lattices but have no capability to rum experiments based on high dimensional datasets. In addition, the experiments show that the more reductions in the attributes dimensionality the more generations of inaccurate clusters appears in the results. These inaccurate clusters are generated due to the loss of important and essential attributes that contain important knowledge related to the studied domains.

4.2.1.2 FCA conclusion

This chapter presented the *FCA* data mining technique for data analysis. The possibility of analyzing and clustering the university graduates' trajectories by applying this technique to low-dimensional dataset is attainable. In our experiments, the *FCA* technique has explored new clusters of university graduates' trajectories based on the dataset collected from our online survey.

This study has revealed that the available *FCA* software and algorithms such as (*ConExp, Conexp-ng-0.7.0, Latviz Loria, Lattice-Miner 2.0 and In-Close*) have many drawbacks: firstly, we found difficulties in converting hundreds of categorical attributes into binary data. Secondly, the available FCA software, do not allow users to control the results and the required number of clusters in their experiments. Finally, the reduction of our dataset attributes from 500 to 119 led to more issues in the experiments and decreased the accuracy of required clusters. These inaccurate clusters are generated due to the loss of important and essential attributes that contain important knowledge related to the studied domains. Therefore, we could not have further reduction in the dimension of the dataset to less than 119 features, because of the existence of many significant attributes that represent useful knowledge related to the graduates' profile, education and career domains. Consequently, we deduced that the *FCA* technique is useful for analyzing low-dimensional datasets, but is not effective in analyzing high dimensional datasets. In the next section, the implementation of the *Hierarchical Clustering* is presented.

4.2.2 Hierarchical clustering

In this section, the implementation of the *Hierarchical clustering* technique is presented. This technique is applied to our dataset in order to cluster university graduates' trajectories. Then, the generated clusters will be used in the RS recommendation process to recommend appropriate recommendations to active students.

The *Hierarchical Clustering* (Murtagh, 2011) is a popular algorithm in machine learning that groups similar objects into clusters. This clustering method compares objects with one another based on their similarity. The importance of this clustering method is that it does not require a preset size of clusters to run the clustering process. *Hierarchical Clustering* was implemented in recommender systems; for example, (Haruechaiyasak et al., 2005) presented a framework for retaining the profiles of the customers in e-commerce RS using the *hierarchical clustering* algorithm. As basic recommender systems have many limitations such as the *Scalability*, the author proposed a dynamic method of retaining customer profiles.

Moreover, this study (Lokhande and Jain, 2019) proposed an approach that implements the *Hierarchical Clustering* algorithm with the CF recommendation process. Additionally, the *Principle Component Analysis* (PCA) technique is used to reduce the dimensionality of the dataset and generate accurate results. PCA is a dimensionality reduction technique that is used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. By incorporating the *Hierarchical Clustering* and the *PCA* techniques with the CF recommendation process, the main elements can be enhanced for the recommendations.

The *Hierarchical Clustering* algorithm runs well with smaller datasets and requires a distance metric such as the *Gower* distance (Gower, 1971) to measure the similarity between objects.

4.2.2.1 The application of Gower distance metric

*T*o measure the nearness or similarity of the objects in a dataset, distance is used to compute the numerical measurement. The *Gower* distance for dissimilarity measures is implemented in the *Hierarchical Clustering* technique, which computes and clusters heterogeneous and categorical data. The *Gower* distance is implemented to measure the dissimilarity between two records whose attribute contains mixed numerical, text, or categorical values. The returns of the distance could be a number between (0) and (1). The value (0) in the distance represents identical objects and value 1 represents the maximally dissimilar. In our case, the *Gower* distance is used to analyze the dissimilarity between the university graduates' records found in our dataset, and then clusters these records based on the dissimilarity matrix.

The used Gower's distance metrics are divided into three types*:* (1) *"quantitative"* that implement range-normalized *Manhattan distance,* (2) *"ordinal"* where variables are first ranked, and then the *Manhattan distance* is implemented with adjustment and (3) *"nominal"* where variables of *k* categories are converted into *k* binary columns.

The Gower dissimilarity equation details:

$$d(i,j) = \frac{\sum_{c=1}^{n} \omega_c \delta_{ij}^{(c)} d_{ij}^{(c)}}{\sum_{c=1}^{n} \omega_c \delta_{ij}^{(c)}}$$

$d(i,j)$ = dissimilarity between row i and row j

c = the cth column

n = number of columns in the dataset

ω_c = weight of cth column $= \frac{1}{\text{nrows in dataset}}$

$$\delta_{ij}^{c} = \begin{cases} 0 & \text{if column c is missing in row i or j} \\ 0 & \text{if column c is asymmetric binary and both values in row i and j are 0} \\ 1 & \text{otherwise} \end{cases}$$

$$d_{ij}^{c}(\text{categorical}) = \begin{cases} 0 & \text{if i and j are equal in column c} \\ 1 & \text{otherwise} \end{cases}$$

$$d_{ij}^{c}(\text{continuous/ordinal}) = \frac{|\text{row i in column c - row j in column c}|}{\max(\text{column c}) - \min(\text{column c})}$$

Equation 4-1

The example below on patients details more this distance.

The dataset sample shown in Figure 4.4 represents 5000 patients records with maximum age is 95 and minimum age is 9 years old. Also, the "GCS" attribute has ranking data ranged from 3 to 15. The value (3) represents "being death" and (15) represents "being most conscious". To calculate the dissimilarity measure between record 1 and 2, we implemented the equation 4.1 and the results are shown in Figure 4.5.

	age	GSC	race	susInfectionFLG
1	45	13	Asian	Y
2	75	7	White	Y
3	62	9	Unknown	Y
4	37	14	Alaska Native	N

Figure 4.4: Patients dataset sample

$$d(1,2) = \frac{\frac{1}{5000} * 1 * \frac{|45-75|}{95-9} + \frac{1}{5000} * 1 * \frac{|13-7|}{15-3} + \frac{1}{5000} * 1 * 0 + \frac{1}{5000} * 1 * 1}{\frac{1}{5000} * 1 + \frac{1}{5000} * 1 + \frac{1}{5000} * 1 + \frac{1}{5000} * 1}$$

$$= \frac{\frac{1}{5000}(\frac{|45-75|}{95-9} + \frac{|13-7|}{15-3} + 1)}{\frac{1}{5000}(4)}$$

$$= 0.4622093$$

Figure 4.5: Dissimilarity distance calculation

The returns from the *Gower distance* are 12,500,000 dissimilarity scores for the 5.000 records. These scores are used in the *Hierarchical Clustering* algorithm in order to cluster patients into similar segments. The implementation process of the *Gower* distance in R through the *daisy* () function involves three main steps:

***First step*:** The Gower distance analyzes the dissimilarity between the records in the dataset.

***Second step*:** Here, the dissimilarity matrix is generated by computing the data frame through the implementation of the *daisy ()* function.

daisy(x, metric = c("euclidean", "manhattan", "gower"), stand = FALSE)

- *x*: The *x* could be a numeric matrix or data frame. The dissimilarities between the rows of *x* are computed by the daisy function. When *x* is a data frame, the columns of class *factor* are measured as nominal and columns of class *ordered* are considered as ordinal variables.
- *metric*: The possible options for *metric* are *euclidean, manhattan, and gower*. When columns of *x* are not numeric values, the "gower" option is chosen.
- *stand*: The measurements in *x* are standardized before calculating the dissimilarities when *stand* is set to "TRUE". To standardize the measurements for each column, the following operation is calculated: (subtracting the variable's mean value and dividing by the variable's mean absolute deviation).

How to apply *daisy()* to the dataset shown in the following figure and generate the dissimilarity matrix?

	Age(N)	Dep	Height(N)	Salary(N)	has_children	CIVIL_STATUS
[1]	22	1	3	0.39	TRUE	MARRIED
[2]	33	3	1	0.34	TRUE	SINGLE
[3]	52	1	2	0.51	FALSE	MARRIED
[4]	46	6	3	0.63	TRUE	DIVORCED
Range	30	NA	2	0.29	NA	NA

Figure 4.6: Personal information dataset

The above figure illustrate the dataset that has four data items where each item is a person. There are six elements: age, department, height, salary, has_childern, and civil_satus. The elements age, height, and salary are numeric. Elements department, has_childern and civil_status are non-numeric. The distance between the first person and second person is 0.590, calculated as follow:

```
       Age Dep  Ht  SAL    HAS_CHILDREN   CIVIL_STATUS
[1] = (22,  1,  3,  0.39,  True,          Married)
[2] = (33,  3,  1,  0.34,  True,          Single)
```

- numeric: abs(difference) / range
- non-numeric: 0 if equal and 1 if different

dist([1], [2]) =

```
   -  Age:            abs((22 - 33) / 30)       = 0.367
   -  Dep:            (different)               = 1
   -  Height:         abs((3 - 1) / 2)          = 1.000
   -  Sal:            abs((0.39 - 0.34) / 0.29) = 0.172
   -  Has_children:   (same)                    = 0
   -  Civil_status:   (different)               = 1

 = (0.367 + 1 + 1.000 + 0.172 + 0 + 1) / 6
 = 3.539 / 6
 = 0.590
```

The Gower distance will always be between 0 and 1. Where a distance of 0 means the two items are the same and a distance of 1 means the two items are as far apart as possible. Thus the computation process of the *daisy()* function will generate a dissimilarity matrix. The Dissimilarity matrix is a mathematical expression of how distant, the item data points in a dataset are from each other, and then cluster the nearest ones together or separate the furthest ones. Therefore, the dissimilarity matrix describes pairwise distinction between items.

Final step: Here the clustering process starts its computation based on the dissimilarity matrix generated in step 2. In this step, the R's *Partition Around Medoids* (PAM) algorithm is used, the size of the cluster is defined by the fast *hierarchical clustering*, and clusters are visualized by the dendrogram.

We applied this clustering technique to the following graduates' job interests attribute shown in Table 4-5.

Graduate Id	Job interests
1	Agriculture, Food, Natural Resources
2	Architecture, Interior Designing
3	Banking, Financial Services, Accounting
4	Business, Management, Administration, Operations
5	Fashion, Textile Designing
6	Architecture, Interior Designing

7	Agriculture, Food, Natural Resources
8	Consultant
9	Business, Management, Administration, Operations

Table 4-5: Job interests clustering

The dissimilarity matrix were generated, and then the *Hierarchical clustering* technique has clustered the graduates into (4) segments based on their job interests. Cluster 1 included graduate 1 and graduate 7 that have job interest in "*Agriculture, Food, Natural Resources"*. Cluster 2 included graduate 2 and graduate 6 that have job interest in "*Architecture, Interior Designing"*. Cluster 3 included graduate 3, graduate 5, and graduate 8 that have job interest in "*Banking, Financial Services, Accounting", "Fashion, Textile Designing", and "*Consultant". Cluster 4 included graduate 4 and graduate 9 that have job interest in "*Business, Management, Administration, Operations"*. The following table shows the clustering results generated by the hierarchical technique. Cluster 1, Cluster 2, and Cluster 3 have contained graduates that have common job interests. Whereas, Cluster 3 have contained graduates that have three different job interests.

Graduate Id	1	2	3	4	5	6	7	8	9
Cluster 1	X						X		
Cluster 2		X				X			
Cluster 3			X		X			X	
Cluster 4				X					X

Table 4-6: The graduates grouped by cluster Id

Finally, the generated clusters were visualized in a dendrogram shown in the following figure.

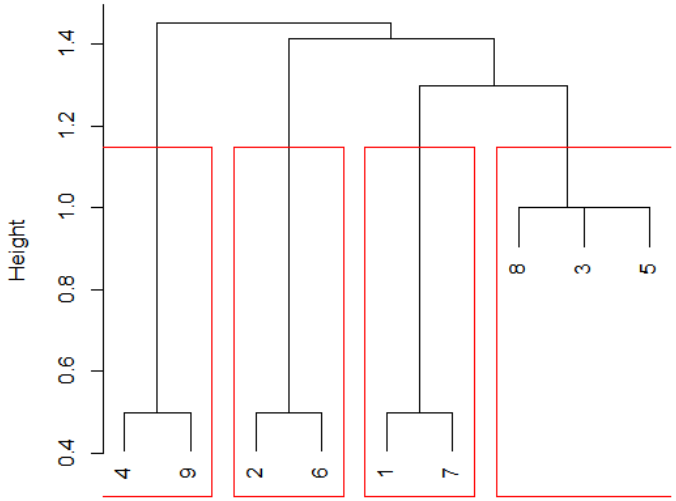


Figure 4.7: The hierarchical clustering illustrated in a dendrogram

Our experiments on the real-world data showed that the *Hierarchical-clustering* algorithm that is based on the *Gower* distance is very efficient in clustering categorical data in low dimensional

datasets. However, our real-world dataset includes about 1000 university graduate's records and approximately 20,000 course ratings which means this dataset has high dimensionality. Additionally, it involved heterogeneous data such as demographic data, interests, characteristics, education and career knowledge, and rating. Besides, the *Hierarchical Clustering* algorithm needs high space and time complexity, in comparison with efficient algorithms, such *k-modes* and *k-Means*. Therefore, it can become hard to determine the number of clusters for high dimensional datasets by the dendrogram. Therefore, we considered this clustering technique inefficient in computing our high dimensional dataset.

4.2.3 K-modes clustering

In this section, the implementation of the *K-modes clustering* technique is presented. This technique is applied to our dataset in order to cluster university graduates' trajectories. Then, the generated clusters will be used in the RS recommendation process to recommend appropriate recommendations to active students.

Since the *Hierarchical Clustering* algorithm needs high space and time complexity in comparison with the k-modes, it can become hard to determine the number of clusters for large datasets by the *dendrogram*. Therefore, this technique is not effective in computing large datasets. Therefore, the *k-modes* algorithm (Huang, 1997) for clustering categorical data is implemented. This algorithm is an extension of *k-means* algorithm, which is very efficient in clustering large datasets. The k-means algorithm computes only numerical values based on the *Euclidean distance*. Whereas, the *k-modes* algorithm is developed to cluster the real-world data containing categorical values. Instead of computing *k-means* distances, the *k-modes* algorithm computes the dissimilarity measure to cluster categorical objects. The total mismatches between two objects is calculated by the dissimilarity measure of the *k-modes* algorithm. The smaller the number of mismatches, the more similar the two objects. This algorithm replaces the means of clusters with modes, defines clusters based on the number of similar categories, and updates modes through a frequency-based method in the clustering process.

- ***Dissimilarity measures***

Consider *m* the categorical attributes, which describe *X* and *Y* as two categorical objects. To define the dissimilarity measure between X and Y, the k-mode algorithm computes the total mismatches of the corresponding attribute categories of X and Y. The smaller the total number of mismatches between the object X and Y, the more similar the two objects.

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$ Equation 4-2

Each category in an attribute is given equal importance by *d(X, Y)*. The dissimilarity measure is defined based on the frequencies of categories in the dataset. Thus, this dissimilarity is defined as

$$d_{\chi^2}(X,Y) = \sum_{j=1}^{m} \frac{(n_{x_j} + n_{y_j})}{n_{x_j} n_{y_j}} \delta(x_j, y_j)$$ Equation 4-3

In this equation the:

- n_{xj}, n_{yj} are the numbers of objects in the dataset that have categories x_j and y_j for attribute j.
- $d_{\chi 2}$ (*X, Y*) resembles the "*chi-square distance"* (Greenacre, 1984). This dissimilarity measure provides more significance to infrequent categories than frequent ones.

The *k-modes* algorithm (Huang, 1997) **consists of four steps*:***

- *Step1:* for each cluster, the algorithm selects one k initial modes.
- *Step 2:* each object is assigned to the cluster whose mode is the nearest to it according to *d*. Then, the mode of the cluster should be updated after each assignment.
- *Step 3:* all objects are assigned to the clusters and a test on the dissimilarity of objects against the current modes is processed. Then, if the nearest mode of an object found belongs to another cluster rather than its recent one, this object should be assigned to that cluster and both clusters should be updated.
- *Step 4:* Iterate step 3 until no change in clusters after a full cycle computation of the entire dataset.

K-modes clustering technique was implemented in recommender systems; for example, this study (Christodoulou et al., 2013) presents the development and implementation of a novel dynamic Web RS using the *K-modes* algorithm. This RS generates recommendations to users based on their interests, prior activities, and behaviors. The recommender process is improved by the use of constant interests that are defined by the user in the registration phase. The RS has been experimented on a movie dataset and the returns show effective recommendations related to users' preferences.

Moreover, this paper (Christodoulou et al., 2017) presented a RS named iBeacons used in a supermarket for suggesting personalized offers to customers. When the customers enter the store,

the iBeacons send them personalized notifications via their mobile devices providing them offers that match their interests. This RS incorporates *Entropy-based algorithm, Bayesian Inference, and k-modes clustering* to generate recommendations to customers related to the available offers. This approach enhanced the shopping experience of the customers by recommending them personalized and accurate suggestions. Additionally, the integrated techniques in this system reduced the problems of basic recommender systems such as *Sparsity, Cold-start,* and *scalability.*

4.2.3.1 The k-modes experiments

In this phase, we implemented many experiments to evaluate the efficiency of *k-modes* algorithm in clustering the graduates' dataset. As an example experiment, we used a dataset of 448 objects (graduates) and 52 variables (attributes). The number of clusters is set randomly to 46 for testing. The k-modes algorithm returned 46 clusters with different sizes of similar graduates. The following sequence represents the number of graduates in each cluster [9, 11, 4, 21, 5, 2, 12, 16, 25, 14, 14, 5, 5, 8, 3, 10, 7, 3, 16, 6, 13, 4, 11, 7, 16, 5, 4, 14, 3, 4, 13, 15, 13, 27, 15, 9, 3, 6, 4, 14, 10, 7, 15, 4, 5, and 11]. The result shows that 46 clusters are too many for clustering 448 objects. Therefore, we set the number of clusters to 10 in order to clusters the 448 objects. The following Table shows the results of the k-mode algorithm based on 10 clusters. Each clusters shows the Ids of similar graduates.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9	Cluster 10
	4	19	7	6	2	5	13	81	15	1
	10	30	9	18	3	11	17	99	20	8
	16	40	12	47	29	24	23	136	21	25
	45	55	14	53	33	31	42	152	22	26
	51	77	34	58	41	32	52	159	28	27
	64	86	38	90	74	43	54	168	36	35
	65	91	49	104	79	56	75	369	44	37
	78	93	57	133	98	59	95	376	48	39
	83	94	61	138	100	63	103	421	50	46
	87	96	66	161	110	68	126		62	60
	129	105	71	164	113	72	130		70	67
G	141	111	106	167	114	80	131		73	69
	169	121	109	172	120	82	157		92	76
R	176	128	146	174	123	84	171		112	85
	205	134	151	177	124	88	185		115	97
A	207	153	160	230	143	89	239		125	102
	216	155	163	232	148	101	266		132	127
D	220	156	170	281	150	107	269		135	137
	250	158	175	305	166	108	272		139	140
U	260	162	178	309	182	116	289		145	144
	271	180	196	314	183	117	291		188	154
A	293	201	198	328	206	118	313		195	165
	307	213	199	330	210	119	316		202	208

T	310	224	219	348	236	122	324		227	209
	335	240	223	349	238	142	331		229	212
E	373	249	226	350	246	147	339		231	214
	374	251	241	355	248	149	343		237	215
	391	294	244	356	256	173	346		242	217
	407	302	254	361	262	179	385		243	218
	444	308	257	367	263	181	400		245	221
	446	317	259	368	274	184	417		247	222
	448	325	261	433	275	186			264	225
		337	265	440	285	187			268	228
I		338	267	442	298	189			278	252
		353	270	447	301	190			279	253
D		364	277		333	191			280	255
		378	283		399	192			282	273
		381	284		422	193			286	276
		382	287		425	194			292	290
		384	297		435	197			295	300
		416	299		437	200			296	318
		426	303			203			311	323
		431	304			204			312	383
		434	319			211			315	387
			329			233			322	395
			341			234			326	424
			354			235			327	
			360			258			332	
			362			288			334	
			365			306			342	
			372			320			344	
			377			321			345	
			380			336			347	
			386			340			351	
			392			352			358	
			393			357			366	
			396			359			371	
			401			363			388	
			402			370			389	
			408			375			394	
			415			379			397	
			420			390			398	
			423			403			404	
			427			405			409	
			436			406			410	
			443			411			412	
						414			413	
						418			419	
						429			428	
						430			439	
						432			441	
						438			445	

Table 4-7: Clusters of graduates' ids

Based on many experiments that we applied to our real-world graduates' dataset, the *k-mode* algorithm proved its efficiency in clustering large categorical datasets. Although, the k-mode algorithm is efficient, it suffers two issues. The first issue is that the solutions are only locally optimal. K-mode just cares about finding the optimal local solution and globally gave incorrect clustering since it focuses on what is good in the neighborhoods and cannot see the big picture. The second issue is that the solutions' qualities are sensitive to the initial conditions (initial choice of centroids). Choosing bad initial centroids gets the k-mode algorithm stuck in bad local optima and running it 5,000 times, and the centroids would not move. The following section recapitulate our work in chapter 4.

4.3 Conclusion

We presented in this chapter three clustering methods namely the FCA, the *Hierarchical Clustering* and the k-modes, in order to recommend personalized university paths to high schools students based on the graduates trajectories. The *k-modes* and *Hierarchical Clustering* algorithm computed approximately equal accuracy when applied to a part of our dataset. However, the *Hierarchical Clustering* could not generate accurate results when applied to the entire dataset. Our analysis showed that the *Hierarchical Clustering* algorithm needs high space and time complexity in comparison with the k-modes. This leads to difficulty in determining the number of clusters for large datasets. Therefore, we considered this technique is not effective for clustering our high dimensional dataset.

Although, the k-mode algorithm is efficient, it suffers two issues. The first issue is that the solutions are only locally optimal. K-mode just cares about finding the optimal local solution and globally gave incorrect clustering since it focuses on what is good in the neighborhoods and cannot see the big picture. The second issue is that the solutions' qualities are sensitive to the initial conditions (initial choice of centroids). Choosing bad initial centroids gets the k-mode algorithm stuck in bad local optima.

As well, our analysis revealed that the available FCA software and algorithms such as (ConExp, Conexp-ng-0.7.0, Latviz Loria, Lattice-Miner 2.0 and In-Close) have many drawbacks: firstly, we found many difficulties in converting hundreds of categorical attributes in our dataset to binary data. Secondly, when using such software, users cannot control the results and the required number of clusters in their experiments. Finally, when we reduced our dataset from 500 to 119 attributes, this reduction led to more issues in our experiments due to the high dimensionality of the dataset. More reduction in our dataset decreased the accuracy of required clusters. Therefore, due to the existence of many significant attributes in our dataset that focused on useful information related to the education domain, we couldn't have further reduced the dimension to less than 119 features.

Therefore, we considered the *FCA* technique useful for low-dimensional datasets but not appropriate for clustering our high dimensional dataset.

The following table illustrates the summary of analysis for the clustering algorithms presented in this chapter.

Clustering Technique	***Dataset type***	***The Technique Efficiency***	***Speed Rate***
K-modes	Nominal	Very efficient in clustering large datasets. However, it suffers two issues: the results are only locally optimal and sensitive to the initial centroids.	Fast
Hierarchical Clustering	Mixed data	Not efficient in processing large datasets	Slow
FCA	Binary	Not applicable to high dimensional datasets	Very Slow

Table 4-8: Clustering techniques analysis

Thus in the next chapters, we present a second approach to recommend a university path to high school students based on CBR and ontology.

CHAPTER 5: Second proposed approach, hybrid RS based on CBR and ontology

This chapter presents the architecture and the development phases of our proposed hybrid RS (COHRS) that is based on the CBR, ontology, KB and CF techniques.

5.1 Introduction

The orientation programs in most schools are not well designed to cater to students' varied needs. In addition to that, the complexity of life and the instability of the job market strongly affects youth when choosing a field of study. Faced with these problems, high school students feel lost when choosing their majors at the university. Besides, the explosive evolution of knowledge and data on the Web network with the growth of innovative electronic machines has made the World Wide Web information increasingly significant in most internet users' life. As a result, internet users are forced to take inappropriate decisions when searching the Web due to an incapability to deal with the massive volumes of data.

Thus, our study focuses on developing a novel hybrid RS that enables students to explore top N recommendations based on their fields of interest. The system general objective is to assist learners in making the right decision when selecting their university/college, university major, and career choices.

As discussed in chapter 4, the FCA clustering technique couldn't cluster the graduates' trajectories due to the high dimensionality of our dataset. Additionally, our analysis revealed the Hierarchical and K-mode algorithms issues and limitations. Therefore, we proposed a different approach based on CBR and ontology. The approach presented in this chapter is a hybrid RS that incorporates two filtering techniques in a uniform system based on the *Feature Augmentation hybrid* strategy. This strategy enabled the recommender engine to incorporate two separate types of recommender algorithms in a way that the output of the first recommender is fed into the input of the second recommender. In addition, this strategy has demonstrated its contribution to improve the performance of the hybrid systems and the quality of recommendations.

COHRS follows 5 steps to generate recommendations to high school students. In step 1, The recommendation process of COHRS starts by enabling high school students to enter their courses' ratings, preferences, interests, demographics data, etc. In this step, students rate 23 high school courses then the system integrates them into the CF recommendation engine. In step 2, the CF generates the recommendation based on high school courses' ratings using the EuclideanDistanceSimilarity metric. In step 3, the CF integrates its output as a new feature into the KB recommender system. In this step, the Feature augmentation hybrid strategy is used to interconnect the KB and CF in a uniform system. In step 4, the KB system processes students' knowledge, queries, and the generated output from the CF to generate the final recommendations. In step 5, the KB system generates personalized recommendations to high school students based on the ontology and CBR system.

The scientific purposes of the proposed hybrid system are described as follows:

- Overcoming the limitations of the traditional recommender systems specifically dealing with high dimensional and heterogeneous data.

- Incorporating the *CF* and *KB* techniques in a uniform system in order to recommend personalized recommendations to the user.
- Integrating *CBR* approach into the *KB* recommender system in order to find similarities between high school student and prior graduates.
- Integrating the ontology into the recommendation process in order to improve the accuracy of the recommendations.

In the next sections, an overview of the proposed COHRS for academic and career guidance is introduced. Also, the architecture detail of this hybrid system that is based on the CBR, ontology, KB and CF techniques is presented. Moreover, the development and phases of the hybrid RS are described.

5.2 COHRS architecture

The proposed approach comprises 3 core layers illustrated in figure 5.1 and described as follows:

The first layer illustrates the hybrid system's GUI. In this layer, the student enters his/her courses' ratings, demographics data, and interests query via the GUI of the system in order to get recommendations. Courses' ratings are integrated into the user-based CF system whereas the demographics data and queries are integerated into the KB system.

The second layer illustrates the hybrid RS, which incorporates the KB system and the user-based CF system. The CF system role is to compute the k-most similarity between the high school student and university graduate and generate recommendations. Whereas the KB role is to generate the overall personalized recommendations to the high school student based on the ontology and CBR system. COHRS implements the *Feature Augmentation* hybridization strategy that enables the CF technique to integrate its recommendations as new features into the KB recommendation process. Then, the KB system matches the similarities between the high school student query and university graduates' cases based on the ontology and CBR system. Finally, the similarity results permit the Hybdrid KB recommender system to generate adequate and personalized recommendations such as a university, univeristy major, and career domain.

The third layer illustrates the domain knowledge, which integrates the concepts and individuals of higher education, school, career, and student profile. The domain knowledge is formally represented in an ontology. In addition, it illustrates the data that are related to the high school students' profiles, ratings, and queries, and university graduates' ratings.

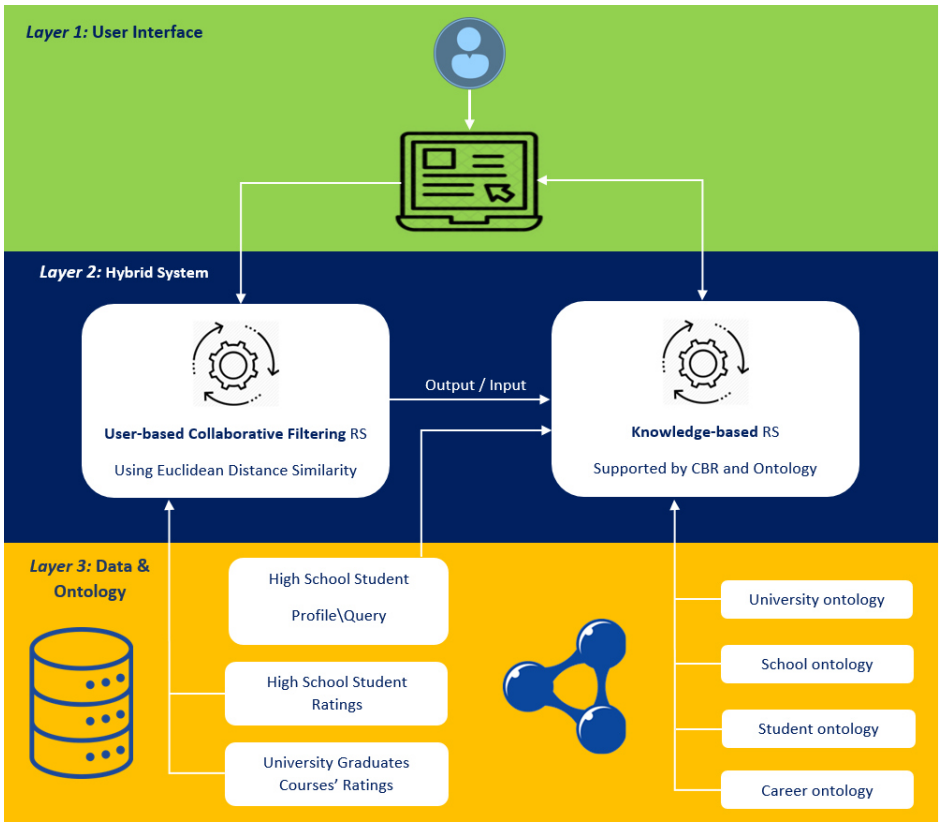


*Figure 5.1: The Architecture of the proposed hybrid recommender system (*COHRS*)*

These three layers interconnect to generate recommendations based on the high school student's preference, interest, demographic data, ratings, and domain knowledge.

5.3 COHRS phases

This hybrid novel approach has involved four main phases:

- The data acquisition from an online survey.
- The data preprocessing by applying many advanced machine-learning techniques to refine and clean the required dataset.
- The ontology design to represent the domain knowledge.
- The mechanism of COHRS engine that recommends personalized recommendations to high school students.

Figure 5.2: Phases of COHRS

5.3.1 Data acquisition phase

This phase represents the explicitly acquired data from our online survey. More details about the online survey are discussed in chapter 4 section 4.2. This survey included more than 50 attributes related to university graduates education and career paths. The collected dataset includes real-world data, of about 1000 university graduate profiles. This dataset consists of domain knowledge, demographic data, interests, and ratings. The demographic data and domain knowledge are integrated into the proposed KB recommendation process in order to overcome the limitations of traditional recommender systems, while ratings are integrated into the CF recommendation process.

5.3.2 Data preprocessing phase

In this phase, the acquired data from phase (1) are transformed into specific formats that fit the recommendation engine of the user-based CF and KB systems. Many advanced data analysis techniques were implemented in this phase such as the *Levenshtein distance* to correct misspelled inputs, and the *InfoGainAttributeEval* Weka tool to rank the dataset's attributes. More details about the data preprocessing analysis are discussed in chapter 4 section 4.3.

5.3.3 The ontology design phase

This phase represents the ontology that provides a semantic description of the education and career domains. CBR recommender systems take advantage of these domains knowledge to obtain accurate results. To design the ontology structure, the "101 methodology" (Noy and McGuinness, 2001) was implemented. The higher education, school, careers, student's profile, and graduate's knowledge were modeled in the ontology using the Protégé OWL Editor. The classes, subclasses, object properties, relations, and individuals were also designed in this part. Our ontology is integrated into the KB recommender system in order to increase the accuracy of the recommendations. Our ontology design encompasses two main segments: the first segment represent the high school student model and the second one represents the *Graduates'* cases that describes all graduates' instances in the knowledge-base such as (graduate's career interests, preferred courses, country, hobby, etc). A graduate is a student who has already a diploma and a job. Figure 5.3 illustrates the graph of our ontology. This figure represents the depth of the subclass hierarchy, which aids in the computation of the similarity measure.

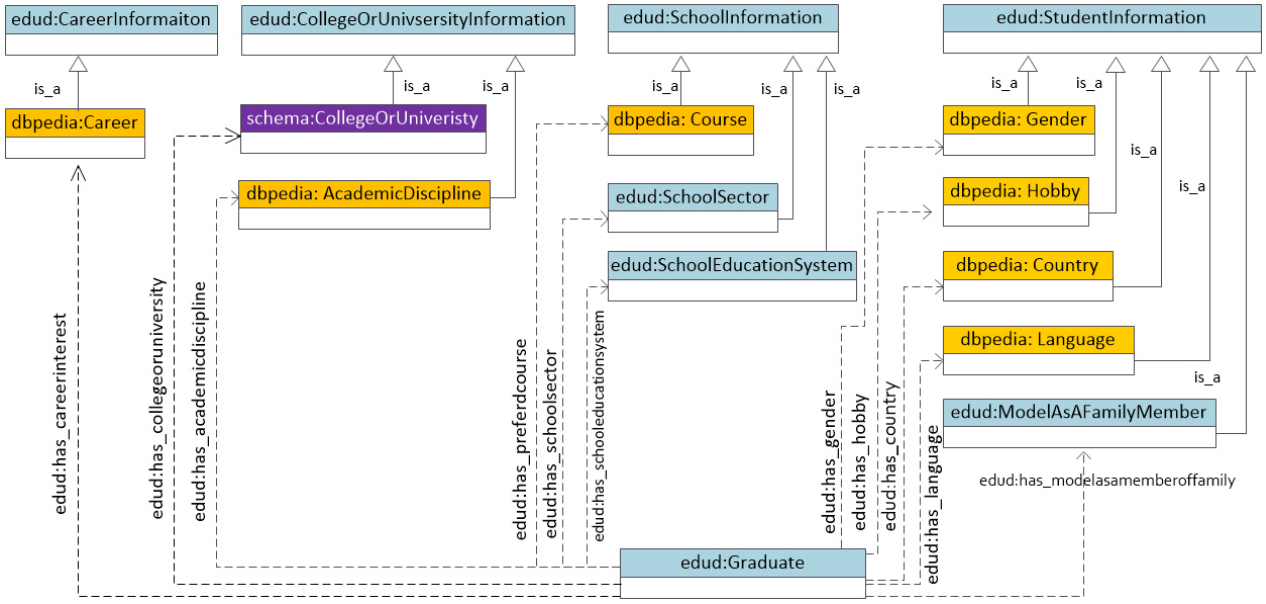


Figure 5.3: *The Graph of the ontology design*

The blue classes and subclasses shown in the above figure represent our ontology concepts, the orange classes and subclasses represents the dbpedia ontology concepts, and the purple class represents the schema ontology concept.
In addition, the *edud* prefix shown in this figure is used as a reference to our ontology concepts.
This ontology design includes five main classes and many subclasses represented as follows:

- CareerInformation class with its subclass Career.
- CollegeOrUniversityInformation class with its subclasses Academicdiscipline and CollegeOrUniveristy.
- SchoolInformation class with its subclasses SchoolSector, SchoolEducaitonSystem and Course.
- StudentInformation class with its subclasses Gender, Language, Country, Hobby, and MondelAsAFamilyMember.
- Graduate class with its object properties has_careerinterest, has_academicdiscipline, has_collegeoruniversity, has_schoolsector, etc.

Each graduate is considered as a case by the CBR system. All cases are embedded into the ontology. This representation allows similarity computation based on distance measures in the ontology. Thus, this technique computes the semantic similarity based on the conceptual model structure and location of concepts in the ontology. This similarity technique takes into account the number of super-classes in the ontology to compute the similarity between two vectors or sets. The following figure illustrates the "*graduate1 case"* with its objects properties linked to its instances in the appropriate subclasses.

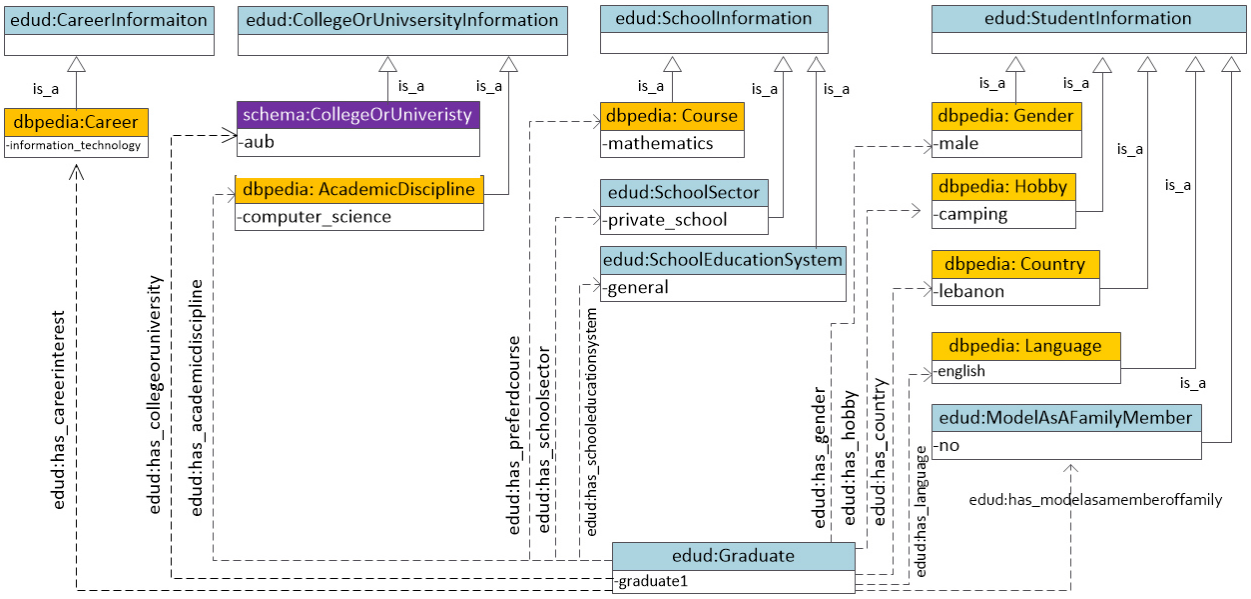


Figure 5.4: Graduate 1 case sample

Graduate1case represents a university graduate that has interests, demographic data and domain knowledge such as:

- has_careerinterest “information_technology”
- has_academicdiscipline “computer_science”
- has_collegeoruniversity “aub”
- has_schoolsector “private_school”
- hs schooleducationsystem “general”
- has_preferedcourse “mathematics”
- has_gender “male”
- has_language “english”
- has_country “lebanon”
- has_hobby “camping”
- has_modelasafamily member “no”

The following figure illustrates the description, solution and graduate instance of a case in the ontology.

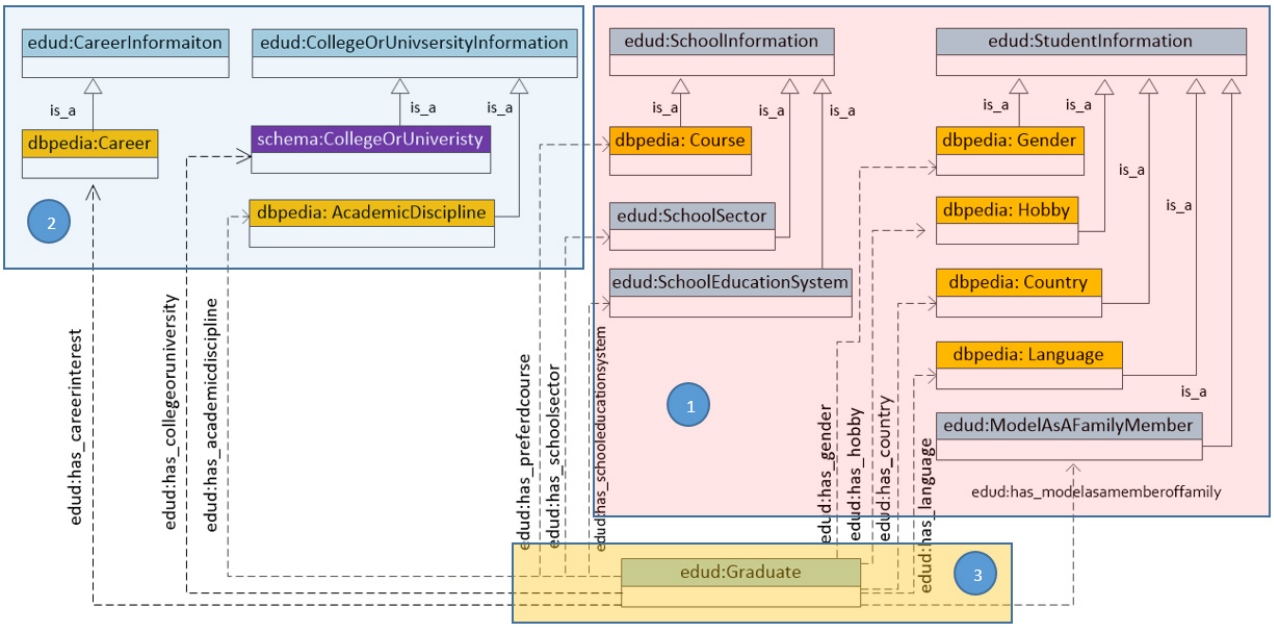


Figure 5.5: Description, solution and graduate instance case

Figure 5.5 shows the representation of a university graduate case that encompasses 3 main sections. Section 1 describes the ***description*** of the case in the ontology; section 2 describes the ***solution*** of the case in the ontology and finally section 3 describes the graduate ***case*** instance in the ontology.

Besides, the following figure illustrates the object properties that describes the problem and solution of a graduate case.

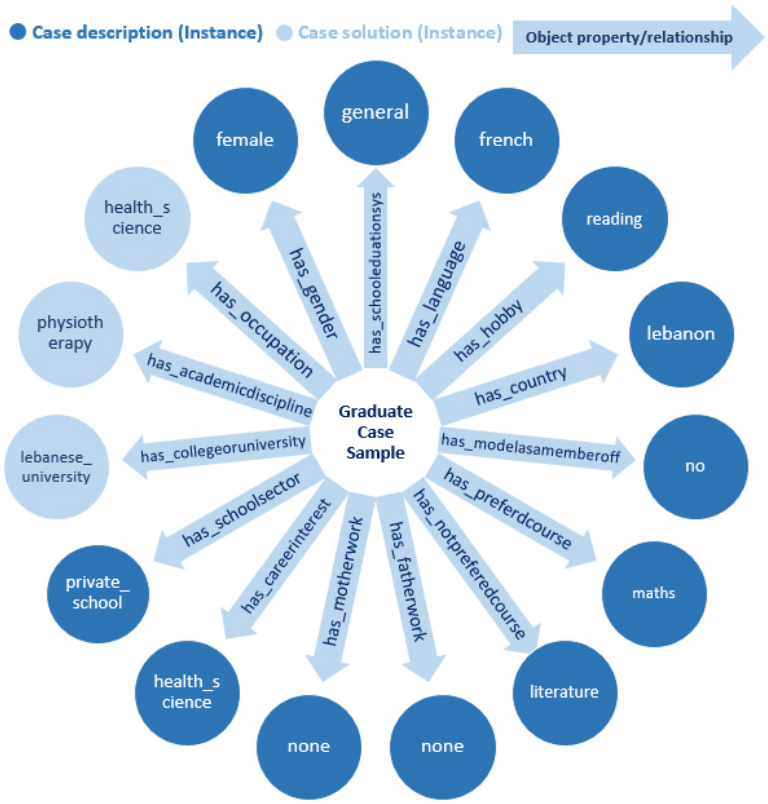


Figure 5.6: A Graduate case consists of a description and solution

In Figure 5.6, the dark blue circles represents the case description, the light blue circles represents the case solution, and the light blue arrows represents the case objects properties in the ontology design. The following figure illustrates the connection between the ontology and CBR system.

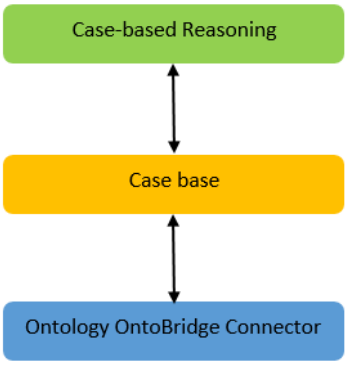


Figure 5.7: Connection between the ontology and CBR

In our case, the complete case base is stored into the ontology. Thus, to use our ontology in the CBR system, the OntoBridge (Recio-García, J. A. et al., 2014) connector is initialized. This connector loads cases represented as concepts or individuals in an ontology. OntoBridge provides a simple wrapper for JENA and allows connecting to PELLET reasoner.

In our work, the ontology plays a significant role in the KB system since it works as:

- A vocabulary to describe domain knowledge.
- Source of knowledge permitting the semantic reasoning in the functions of similarity computation.
- A conceptual model structure where the cases are located.

- **Integrating the graduate cases as instances into the ontology**

In our study the analysis are based on the university graduates' cases that were extracted from our survey. These prior graduates' cases were transformed into instances and stored in the ontology design. The final refined case base encompasses 658 graduate cases that represent only the university graduates that have a university major related to their current job and their job meets their interests. In order to integrate the graduate cases as ontology instances into the KB system, we created an algorithm that does the following:

1. *Read all graduate cases from an excel sheet.*
2. *Add an object property to each attribute in the case.*
3. *Transform the case to an ontology instance.*
4. *Save the cases in an RDF file.*
5. *Integrate the RDF file into the KB engine.*

The following figure illustrates an ontology instance sample generated automatically in our system from a graduate case and saved in an RDF file.

```
<!-- http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#Graduate2 -->
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#Graduate2">
<rdf:type rdf:resource="http://dbpedia.org/resource/Graduate"/>
<has_gender rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#female"/>
<has_country rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#lebanon"/>
<has_language rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#french"/>
<has_hobby rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#crafts"/>
<has_modelasamemberofthefamily rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#yes"/>
<has_schoolsector rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#private_school"/>
<has_schooleducationsystem rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#general"/>
<has_course rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#mathematics"/>
<has_notpreferedcourse rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#history"/>
<has_careerinterest rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-
265#arts_audio/video_technology_and_communications"/>
<has_fatherwork rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#education_and_training"/>
<has_motherwork rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#education_and_training"/>
<has_academicdiscipline rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#physiotherapy"/>
<has_collegeoruniversity rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#usj"/>
<has_occupation rdf:resource="http://www.semanticweb.org/user/ontologies/2021/1/untitled-ontology-265#health_science"/>
</owl:NamedIndividual>
```

Figure 5.8: An ontology instance that represents a graduate case

- **Importing large number of individuals into the ontology**

Besides, in order to import large number of individuals into the ontology design, we used the *cellfie*[7] *protégé* plugin. *Cellfie* supports translating axioms from Excel workbooks.
The following are some *cellfie* rules that helped us the import instances into the ontology design:

- *Individual: @A* Types: schema:CollegeOrUniversity*
- *Individual: @B* Types: dbpedia:AcademicDiscipline*
- *Individual: @C* Types: dbpedia:Career*
- *Individual: @D* Types: dbpedia:Course*
- *Individual: @E* Types: dbpedia:Hobby*

The following figure illustrates an excel sheet with column "A" that represents the university name and the transformation rule of the *cellfie* plugin that transforms column "A" to an ontology instance.

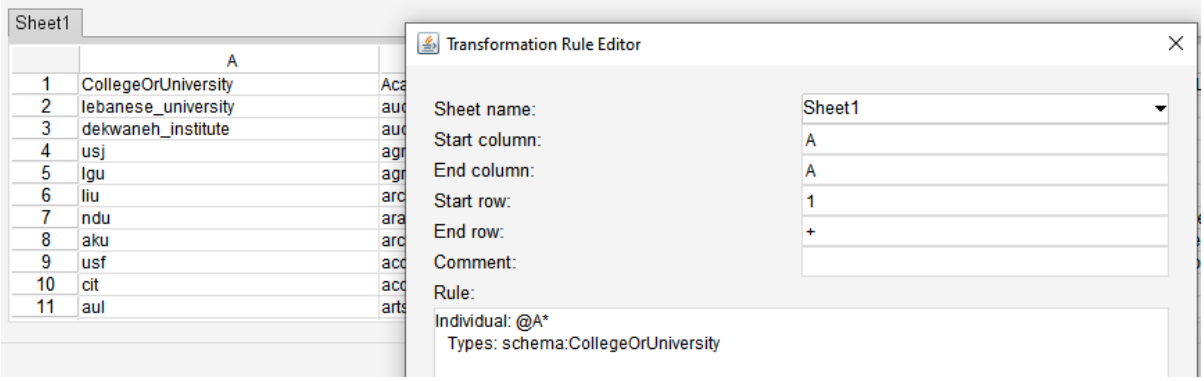


Figure 5.9: The Cellfie rule example

- **upgrading the old jColibri query interface and replace it by our query interface**

Figure 5.11 represents a high school student's query sample. This query includes the domain knowledge, students' interests and preferences, and demographic data. At the beginning of our work we used the existing jColibri query interface that uses the ontolgy explorer to build the query from the ontology, as illustrated in the below figure.

[7] Cellfie (https://github.com/protegeproject/cellfie-plugin) allows to import spreadsheets content inside OWL ontologies in Protégé

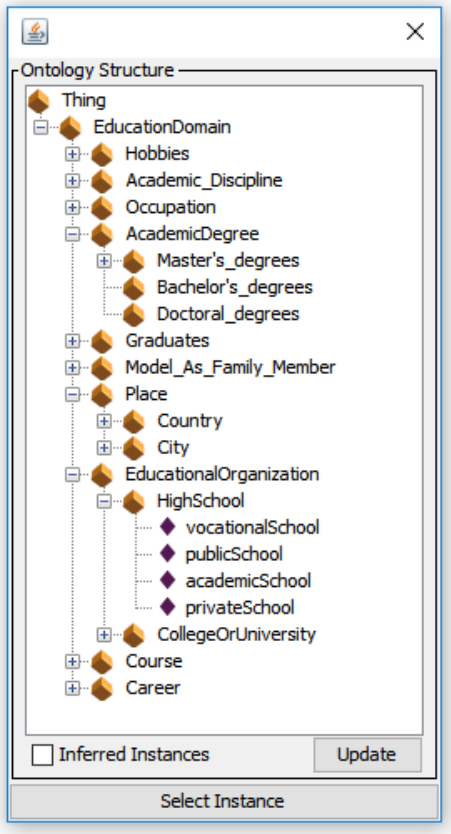


Figure 5.10: The old jColibri query interface

However, we found that the jColibri query interface is hard to use specially in high dimensional ontology that contains high number of instances. For instance, when the user builds his/her query, he/she should select an instance from the jColibri ontology interface that is shown in Figure 5.10. To select the required instance, the user should collapse every class and subclass to explore the available instances and select the appropriate feature. Thus, we developed and upgraded the jColibri query interface and replace it by our query interface that is illustrated in the following figure. Our query interface is based on list boxes that simplify the features selection.

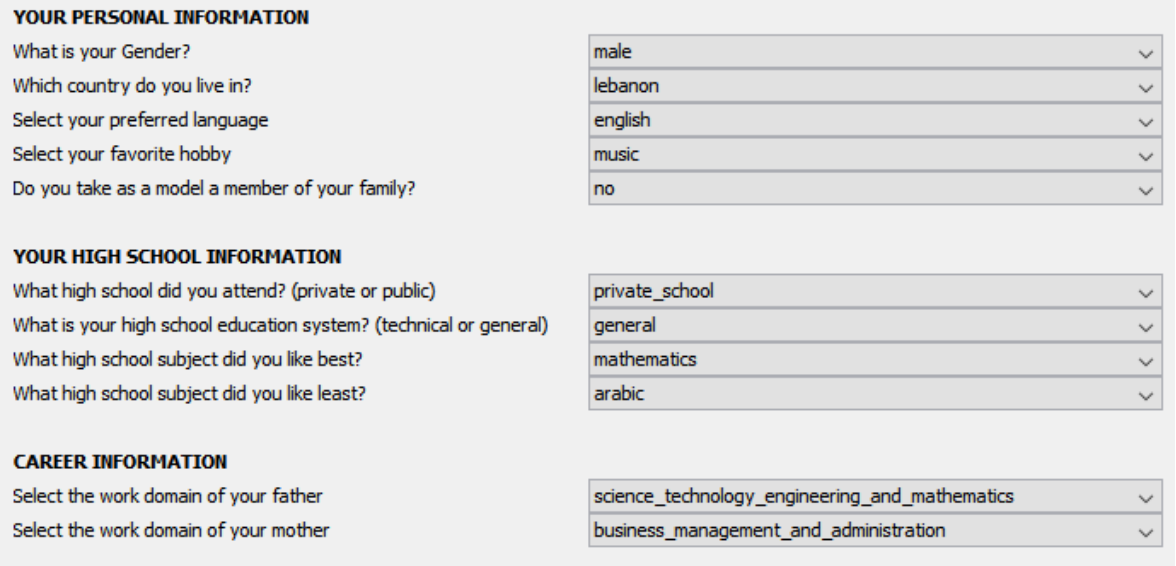


Figure 5.11: Query sample in our approach number 5

- ***Computing similarities using ontologies***

In this section, we represent the *detail* (i_1, i_2) equation of the jColibri2 (Recio-García, J. A. et al., 2014) java library, which computes the semantic similarity based on the ontology structure and location of concepts in it. To find the similarity between concepts, the ontology is needed to compute the semantic relationship. In addition, the conceptual model of the ontology permits the reasoning at all concept levels. Thus, to overcome the limitations of the traditional RS, the ontology is integrated to represent the domain knowledge and compute the semantic similarity between the concepts. Our ontology represents the knowledge of education domain, career domain, students' profiles and interests, and university graduates' prior cases.

The jColibri2 *detail* equation computes the distance in the ontology between the query attribute and retrieved case-matching attribute. This similarity function takes into account the number of super-classes in the ontology to compute the similarity between two vectors or sets.
The jColibri2 *detail* equation computes the similarity by implementing the method *compute(caseObject, queryObject)*, which computes the similarity between two concepts. Where *caseObject* represents the concept of the case and the *queryObject* represents the concept of the query. The below represents the *detail* (i_1, i_2) *equation* of the jCollibri2 library.

$$\text{detail}(i_1, i_2) = \text{detail}(\text{t}(i_1), \text{t}(i_2)) = 1 - \frac{1}{2 \cdot \left| \left(\bigcup_{d_i \in \text{t}(i_1)} (\text{super}(d_i, CN)) \right) \cap \left(\bigcup_{d_i \in \text{t}(i_2)} (\text{super}(d_i, CN)) \right) \right|}$$

Equation 5-1

- i_1 is the concept of the active student query.

- i_2 is the concept of the prior graduate case.
- *CN* is the set of all the concepts in the domain ontology.
- *super* (d_i, *CN*) is the subset of concepts in *CN* which are super concepts of d_i.
- *super* (d_{i1}, *CN*) ∩ *super* (d_{i2}, *CN*) represents the intersection size, which is the common/shared super concepts between the compared student query and prior graduate case.

The following figure shows an example of a hierarchy tree of the ontology structure.

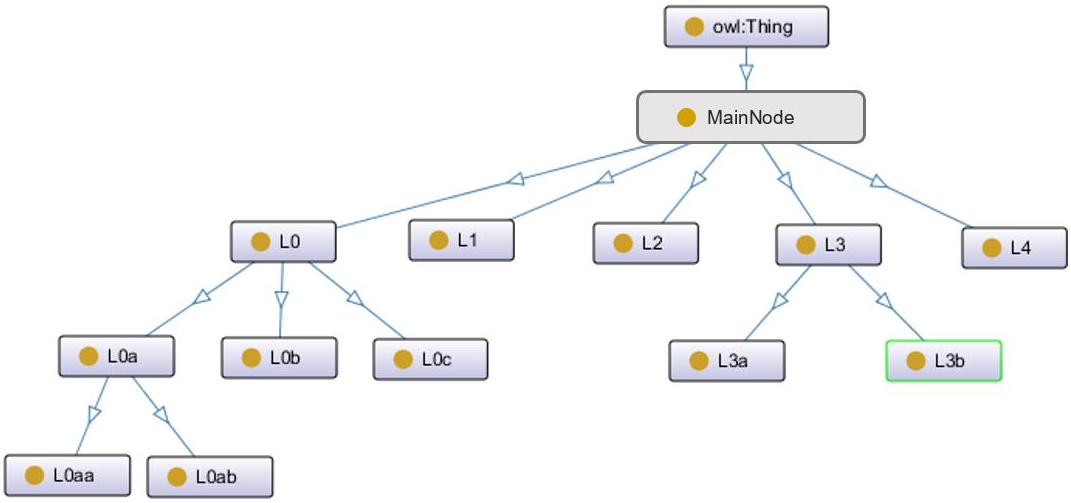


Figure 5.12: Example of a hierarchy tree of an ontology

Here we present the *Detail* equation for similarity rate applied on the ontology structure shown in Figure 5.12. In this equation, super(d_i, CN) consider the amount of common/shared super-concepts to measure the similarity distance. Thus, higher the amount of common/shared super-concepts, shorter the similarity distance between the two compared concepts. Accordingly, super (L0aa, L0ab) from Figure 5.12 will consider three supper-concepts the "L0a", "L0" and "MainNode", while super (L0, L3) will consider one super-concepts the "MainNode" to compute the similarity distance. Thus, in this example super (L0aa, L0ab) will compute lower similarity distance than super (L0, L3). The following table shows the shared super-concepts between two concepts X and Y.

Concept X	***Concept Y***	***Shared Concepts (path to MainNode)***	***Similarity Between X & Y***
L0aa	L0ab	L0a, L0, MainNode	High
L0c	L0b	L0, MainNode	Average
L0c	L3a	MainNode	Low

Table 5-1: Semantic similarity calculation between two concepts

We applied this method to calculate the similarity between 2 concepts in our ontology. The following figure represents the depth of the subclass hierarchy of our ontology design.

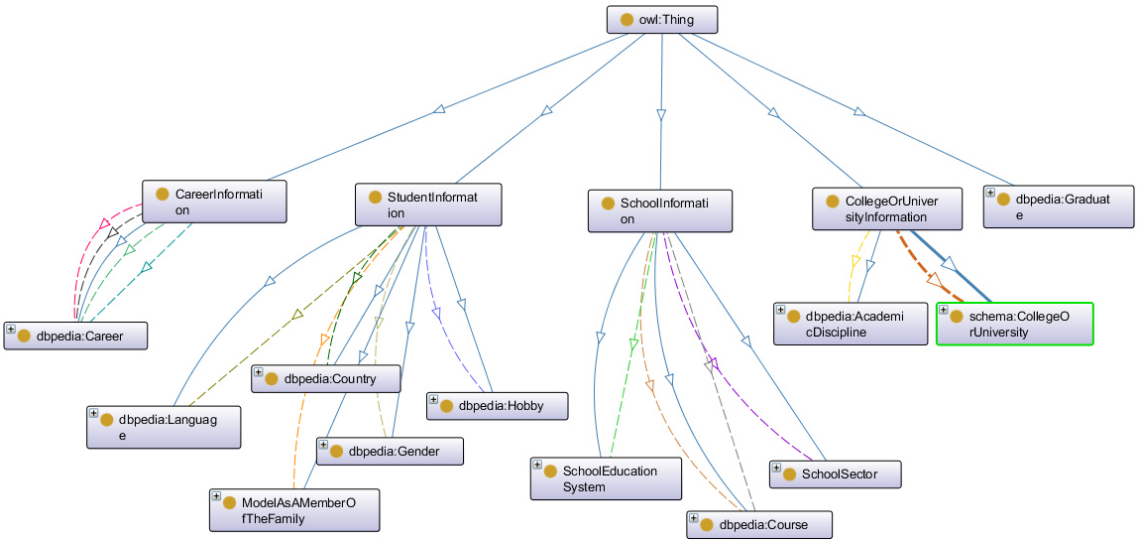


Figure 5.13: The graph of the ontology

The following section presents COHRS recommendation engine phase.

5.3.4 COHRS recommendation engine phase

Once the data and ontology are prepared, the RS computes the similarities and provides recommendations to the high school student. The engine of this hybrid RS integrates two core recommender systems namely the KB and CF that are illustrated in Figure 5.1.

The main function of the user-based CF system is to compute the high school students and university graduates' ratings and find interests similarities in order to generate appropriate career recommendation. Then the output from the user-based CF recommender system is integrated as a new feature into the KB recommender system in order to recommend the final personalized recommendations. In the KB system, the semantic similarity is computed through the ontology structure based on the hierarchical order between the ontology concepts.

The user-based CF technique integrates ratings data in order to find interests similarities between high school students and university graduates, and then generates career domain recommendation. The proposed user-based CF recommendation is based on high school and graduates courses' ratings and careers' ratings. The university graduates rated their career domain and their level on 23 high school courses namely the *Literature, Philosophy, Religion, Music, Theatre, Dance, Drawing, Biology, Chemistry, Physics, Mathematics, and Science of engineering, History, Geography, Economics, Sociology, Psychology, Arabic language, English language, French language, other foreign language, Technology and Computer Science,* and *Physical education*. This rating process helped us to collect from the online survey, approximately 20,000 ratings.

The following table shows a high school courses' ratings sample. For instance, these ratings show that the university graduate was poor in music and dance but very good in literature, religion, biology, etc. Thus, the RS will use these ratings to recommend to high school student who has similar ratings a similar career interest.

Course Name	*Course Id*	*Evaluation (Very Good=3, Good=2, Poor=1)*
Literature	1	3
Philosophy	2	2
Religion	3	3
Music	4	1
Theatre	5	2
Dance	6	1
Drawing	7	2
Biology	8	3
Chemistry	9	3
Physics	10	3
Mathematics	11	3
Science_of_engineering	12	3
History	13	2
Geography	14	2
Economics	15	2
Sociology	16	2
Psychology	17	2
Arabic_language	18	3
English_language	19	3
French_language	20	3
Other_foreign_language	21	1
Technology_and_Computer_Science	22	3
Physical_education	23	2

Table 5-2: Student courses' rating sample

In order to conduct the CF experiments in chapter 6, we prepared and refined a dataset that encompasses 469 objects and 39 attributes. These objects represent the university graduates that have a job interest similar to their real job. Besides, the 39 attributes represent their high school courses and careers' ratings. This dataset contains about 11,000 ratings from 469 users on 39 items. All users in the dataset rated at least 20 items. To ensure the accuracy of our experiments' returns, we assumed that the real-world data collected from our survey are correct. The correctness of our dataset is motivated by the way we disseminated and collected the survey entries. This survey was disseminated to real university graduates that study in different disciplines and real employees that work in different domains. In addition, the data collection process involved face-to-face interviews

to fill the intended survey. Since the selected dataset involve only graduates having a job interest similar to their actual job, we considered it a trusted real-world dataset.

The interconnection strategy between the KB and CF recommender systems is based on the Feature Augmentation hybrid strategy. Researchers categorized the RS hybridization into two main cases. The first case is the uniform in which one RS algorithm has better precision than another algorithm over the entire space of recommendation. For instance, the Cascade strategy with the stronger RS given higher priority, the Feature augmentation strategy in which the weaker RS algorithm performs as an assistant contributing a small amount of info, and the Meta-level strategy in which the stronger algorithm generates a heavy representation that reinforces the performance of the weaker algorithm.

The second case is the non-uniform in which two recommender algorithms have different powers in different parts of the space. In this case, the process will need to be able to employ the two-recommender algorithms at different times. For instance, the Switching strategy is a natural choice here and needs the system to detect when one algorithm should be favored. The Mixed and Feature combination strategies permit output from both RS algorithms without applying a switching measure.

In our hybridization strategy, we implemented the *Feature augmentation* technique because our KB system is the stronger algorithm based on the domain knowledge and the CF is the weaker one based on the ratings. This strategy enabled a contributing CF recommender to make a positive effect without interfering with the performance of the KB algorithm. The following figure illustrates the *Feature augmentation* hybrid procedure.

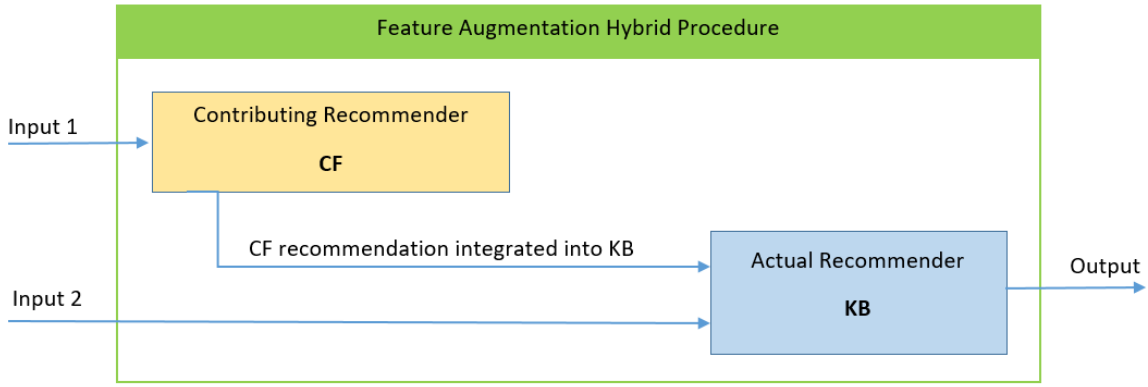


Figure 5.14: Feature augmentation hybrid procedure

The following sequence diagram illustrates *COHRS* recommendation sequence diagram:

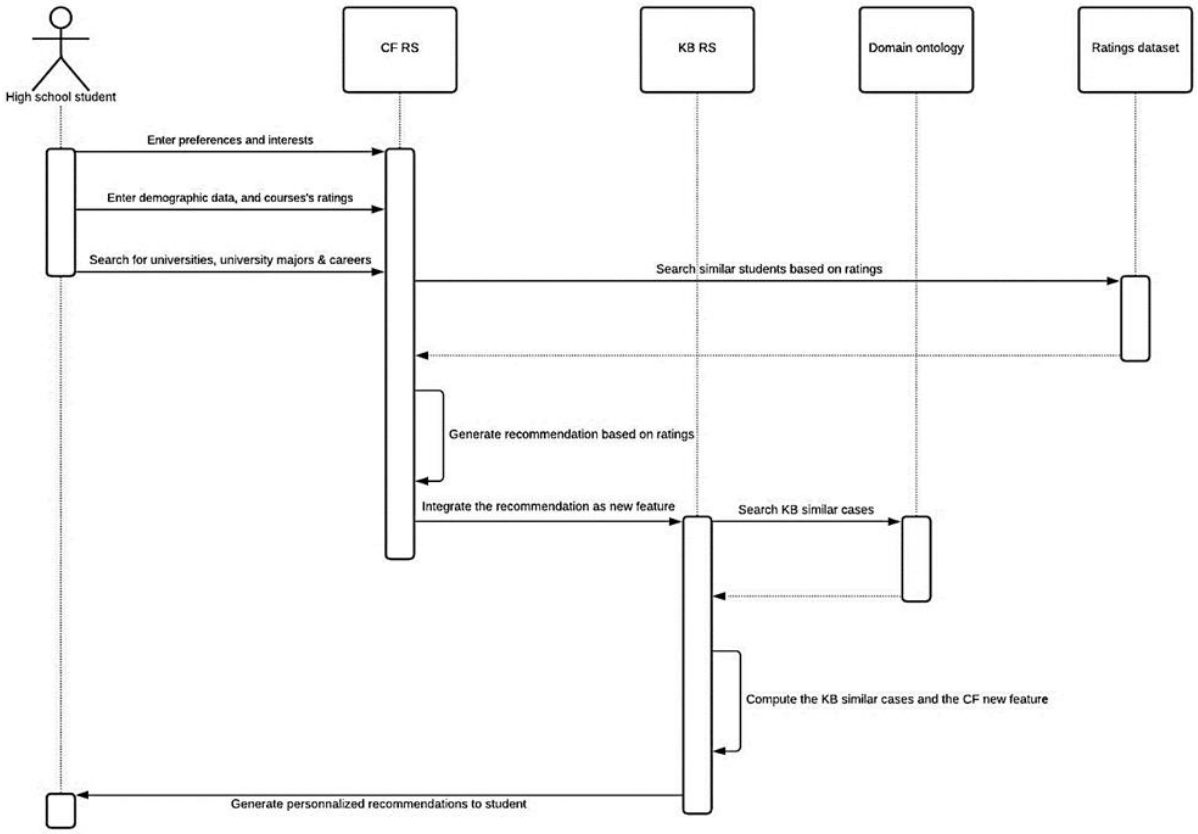


Figure 5.15: The COHRS sequence diagram

The recommendation sequence illustrated in Figure 5.15 starts with step 1 that enables the high school student to enter his/her courses' ratings, preferences, interests, demographic data, etc. into COHRS. At this point, the student will rate 23 high school courses that will be integrated into the CF recommendation engine. In step 2, the CF generates the recommendation based on the high school ratings using the *EuclideanDistanceSimilarity* metric. In step 3, the CF integrates its recommendation as a new feature into the KB recommender system. In this step, the *Feature augmentation hybrid* strategy is used to incorporate the KB and CF in a uniform system. In step 4, the KB system uses the student's knowledge and the recommendation generated by the CF to compute the final recommendations. In step 5, the KB system generates the personalized top N recommendations based on the CBR and ontology.

In order to develop the proposed hybrid RS, the java language was used in the development phase. Thus, we based our development code on many open-source java library such as (Apache Mahout (Giacomelli, 2013), jColibri2 (Recio-García, J. A. et al., 2014), OntoBridge (Recio-García, J. A. et al., 2014), Jena (McBride, 2002), and Pellet (Sirin et al., 2007)).

5.4 Conclusion

In this chapter, an overview of the proposed COHRS for academic and career guidance is introduced. Also, the architecture detail of this hybrid system that is based on the CBR, ontology, KB and CF techniques is presented. Moreover, the development and phases of COHRS are discussed.

The proposed hybrid approach comprises 3 core layers illustrated in figure 5.1. The first layer illustrates the GUI of the hybrid recommender system. In this layer, the student enters his/her courses' ratings, demographics and interests query via the GUI of the system in order to get recommendations. The second layer illustrates the hybrid RS, which incorporates the KB system and the user-based CF system. The CF system role is to compute the k-most similarity between the high school student and university graduate and then generate the recommendation. Whereas the KB role is to generate the final personalized recommendations to the high school student based on the ontology and CBR system. COHRS implements the *Feature Augmentation* hybridization strategy that enables the CF technique to integrate its recommendations as new features into the KB recommendation process. Finally, The third layer illustrates the domain knowledge, which integrates the concepts and individuals of higher education, school, career, and student profile. This domain knowledge is formally represented in an ontology. In addition, it illustrates the data that are related to the high school students' profiles, ratings, and queries, and university graduates' ratings.

Besides, our hybrid novel approach has involved four main phases. The first phase describes the data acquisition process. The second phase describes the data preprocessing work that uses many advanced machine-learning techniques to refine and clean the required dataset. The third phase describe the conceptual model of the ontology that represent the domain knowledge. Finally, the fourth phase that describes the mechanism of COHRS that recommends personalized recommendations to high school students.

In the next chapter, several recommendation techniques and hybridization approaches such as (the stand-alone user-based and item-based CF recommender systems, the stand-alone DF recommender system, the stand-alone KB recommender system supported by CBR, the stand-alone KB recommender system supported by CBR and ontology and the KB Hybrid RS incorporated with the user-based CF and supported by CBR and ontology) are experimented and evaluated.

CHAPTER 6: Comparative analysis and evaluations of recommender systems

This chapter presents the implementation and evaluation of five RS approaches namely the stand-alone user-based and item-based CF recommender systems, stand-alone DF recommender system, stand-alone KB recommender system supported by CBR, stand-alone KB recommender system supported by CBR and ontology, and KB Hybrid RS incorporated with the user-based CF and supported by CBR and ontology.

6.1 Introduction

The hybridization approach has proved its ability to address the limitations of filtering approaches or single-approach in recommender systems. To address these issues, a hybrid RS is recommended. Therefore, in this chapter, we proposed 5 approaches to experiment based on the approach presented in chapter 5. The objective of this experiment is to evaluate and compare them with our approach (chapter 5) to prove the efficiency of the techniques used in our hybrid RS.

In this study, the selection of the 5 below approaches for our experiments and assessments were affected by the data types and high dimensionality of attributes in our dataset. For instance, the ratings were treated by the CF technique whereas the domain knowledge and demographic data were treated by the KB and DF techniques.

Thus, we implemented and evaluated the following five approaches:

(1) The stand-alone user-based and item-based CF recommender systems.
(2) The stand-alone DF recommender system.
(3) The stand-alone KB recommender system supported by CBR.
(4) The stand-alone KB recommender system supported by CBR and ontology.
(5) The KB Hybrid RS incorporated with the user-based CF and supported by CBR and ontology (COHRS).

(1) The **stand-alone user-based and item-based CF recommender systems** process the high school student courses' ratings and recommend him/her appropriate career options. These two techniques compares the active student ratings with the university graduates' ratings in order to match similarities between the users (students) and items (courses). This CF system was experimented basing on five similarity metrics namely the *Euclidean Distance Similarity, Pearson Correlation Coefficient Similarity, Spearman Correlation Coefficient Similarity, City Block Similarity,* and *Uncentered Cosine Similarity.*

(2) The **stand-alone DF recommender system** processes the high school students' demographic data in order to recommend some appropriate recommendations such as university majors, career domains, etc. This system uses' demographic data such as gender, location, language, etc.

(3) The **stand-alone KB recommender system supported by CBR** processes the knowledge about the users to recommend personalized recommendations such as a university, university majors, and career domains. This system compares the high school student knowledge case with the university graduates' knowledge cases in order to retrieve most similar cases from the case-base and then generates top N recommendations.

(4) The **stand-alone KB recommender system supported by CBR and ontology** processes the knowledge of high school students to recommend personalized recommendations such as a university, university majors, and career domains. This system compares high school student knowledge case with the university graduates' knowledge cases in order to retrieve most similar cases from the case-base and then generates top N recommendations. The difference between system (4) and system (3) is in the similarity computation, which is based on the ontology. The knowledge in this system is represented by the ontology and integrated into the KB recommendation process.

(5) The **KB Hybrid RS incorporated with the user-based CF and supported by the CBR and ontology (COHRS)** computes high school students and graduates' ratings, interests, domain knowledge, and demographic data in order to recommend personalized recommendations such as a university, university major, and career domain. This approach is a combination of approaches (1) and (4). This system incorporates four core technologies namely the KB, CF, CBR, and ontology. The CF and KB techniques cooperate in a hybrid system using the "*Feature Augmentation Hybrid*" approach. This hybridization approach enables the CF to integrate its recommendations as new features into the KB recommendation process. Then the KB engine generates the final top N recommendations with the support of the ontology and CBR.

In the next section, we present the experiments, evaluations, results, and comparative analysis of the five mentioned approaches.

6.2 Implementation, results and evaluations

In this section, the implementation, comparative analysis, and evaluation of the five-recommender systems approaches are presented.

6.2.1 The stand-alone user-based CF recommender system

In this section, a stand-alone CF technique is tested and evaluated. This technique integrates ratings data in order to find interests similarities between high school students and university graduates, and then generates career domain recommendation.

The proposed CF recommendation is based on high school and graduates courses' ratings and careers' ratings. The university graduates rated their career domain and their level on 23 high school courses. The following table shows a courses' ratings sample of a high school student. The CF recommender system engine uses these ratings to recommend to high school student a career domain based on the prior university graduates' ratings.

Course Name	*Course Id*	*Evaluation (Very Good=3, Good=2, Poor=1)*
Literature	1	2
Philosophy	2	3
Religion	3	2
Music	4	3
Theatre	5	1
Dance	6	2
Drawing	7	2
Biology	8	1
Chemistry	9	3
Physics	10	3
Mathematics	11	2

Table 6-1: Student courses' rating example

In this section, we implemented and evaluated the User-based and Item-based CF algorithms in order to demonstrate the efficiency of the system based on our ratings dataset. In order to conduct the CF experiments, we used a dataset of 469 objects and 39 attributes. The objects represent the university graduates and the 39 attributes represent their high school courses and careers' ratings. This dataset contains about 11,000 ratings for 39 items provided by the 469 graduates. All the university graduates in the dataset rated at least 20 items.

This experimental study divides the dataset into two sub-datasets. The first sub-dataset contains the training data and the second sub-dataset contains the testing data. For each similarity metrics such as the *Euclidean Distance Similarity, Pearson Correlation Similarity*, *etc.,* evaluation has been implemented based on the MAE and RMSE.

Since the experiment is based on item ratings, we implemented and evaluated the User-based and Item-based CF algorithms based on the *Euclidean Distance Similarity, Pearson Correlation Similarity, Spearman Correlation Similarity, Uncentered Cosine Similarity, and City Block Similarity.* The main function of the mentioned metrics is to find similarities between graduates and high school students based on their ratings. Then, the CF recommender system will recommend a career that is adequate to the high school's interests.

In this experiment, some parameters have been determined such as the *N neighborhood* size, and training *ratio* of the experiment. In addition, the effects of different CF algorithms and similarity metrics were considered. The *N neighborhood* represents the nearest-neighbors to the object location. With user neighborhood, the RS can find the most similar user for the selected user. The training *ratio* represents the percentage of each user's preferences to use to produce recommendations; the rest of ratio are compared to estimated preference values to evaluate recommender performance.

To evaluate our CF recommender system approach, we implemented the mahout evaluation method (Giacomelli, 2013). This evaluation method evaluates the accuracy of recommender systems' recommendations. Applications will take a percentage of the preferences provided by the given DataModel as training data. This is classically most of the data (90 percent). This data is used to produce recommendations, and the rest of the data is compared against estimated preference values to see how much the recommender's predicted preferences match the user's real preferences. Precisely, for each user, this percentage of the user's ratings are used to produce recommendations, and for each user, the remaining preferences are compared against the user's real preferences. The return is a score representing how well the recommender's estimated preferences match real values. Lower scores mean a better match and 0 is a perfect match.

The size of the Neighbor can affect the prediction quality. By changing the number of neighbors, the sensitivity of the neighborhood is determined. In this section, the User-based and Item-based CF algorithms are evaluated and tested based on many similarity metrics and neighborhood sizes.

The result of many experiments shows that the User-based CF algorithm and the Euclidean Similarity metric generated the lowest MAE result that is equal to 0.45 and RMSE result that is equal to 0.58, which means they predict better than the Item-based CF algorithm and its similarity metrics. The experiments of the two CF algorithms' results are illustrated in Figure 6.1 and Figure 6.2.

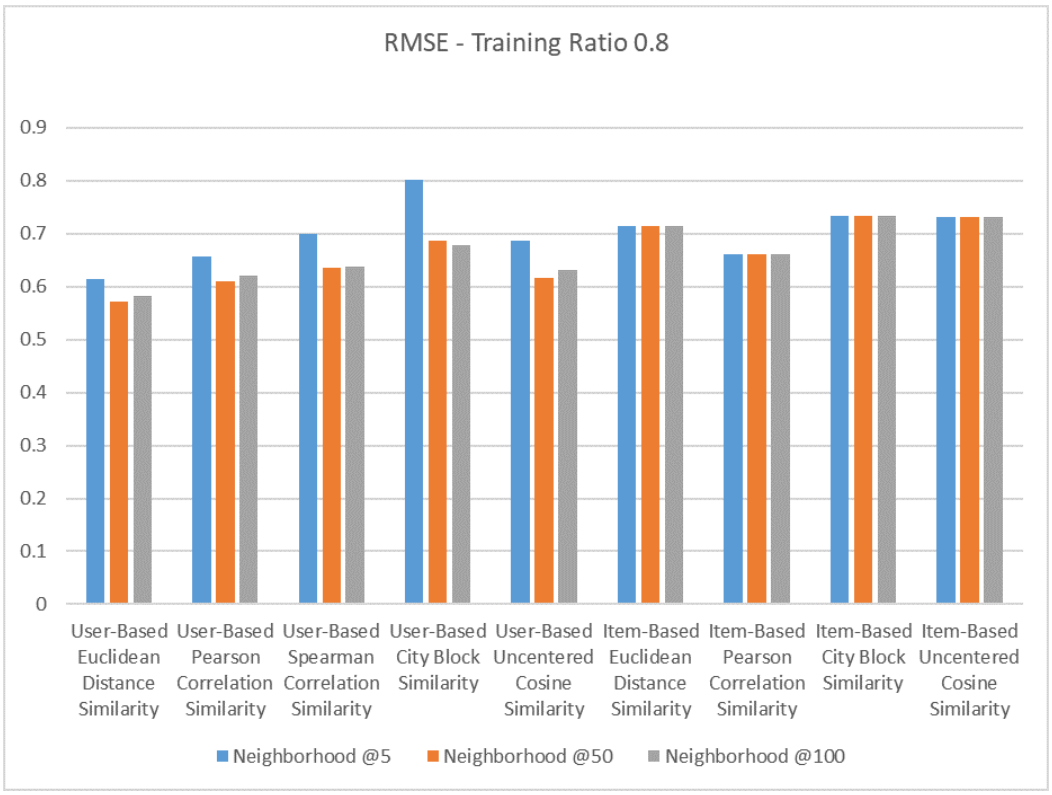


Figure 6.1: RMSE for User-based and Item-based similarities with training ratio equal to 0.8

The following is an example of a recommendation based on our proposed User-based CF approach:

*"**Recommended university major:** Information Technology, similarity rate: 3.0"*

In the above recommendation, the term *similarity rate* represents the highest rate of the recommended item, which means it is 100% similar. Whereas, the *Information Technology* represents the recommended career domain.

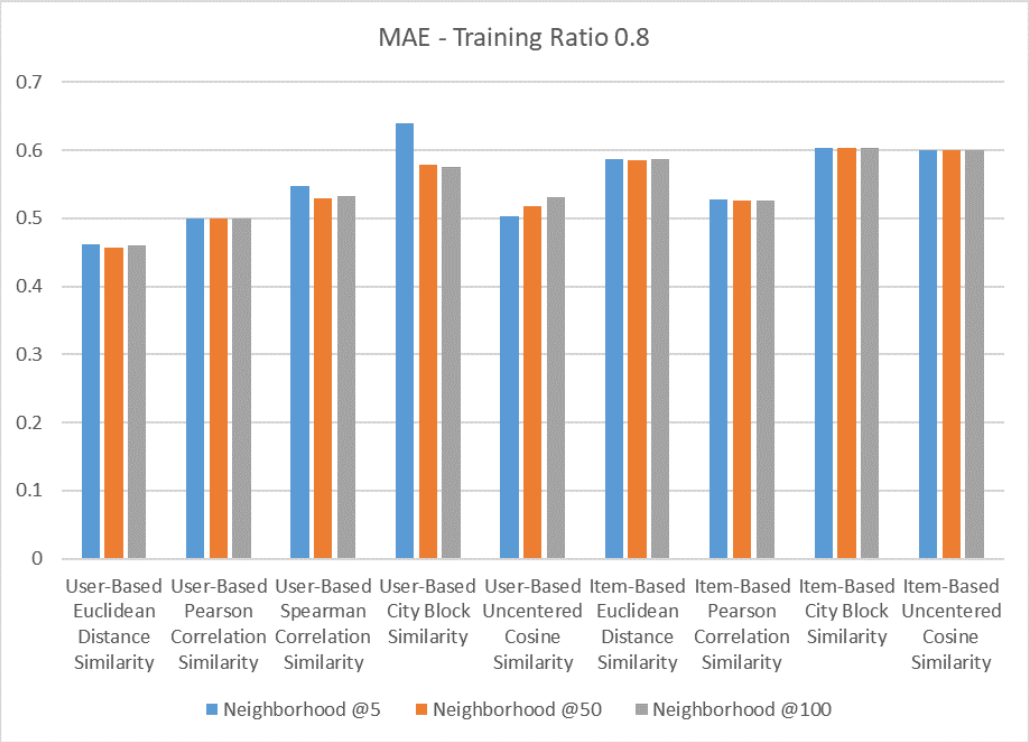


Figure 6.2: MAE for User-based and Item-based similarities with training ratio equal to 0.8

All our experiments showed that the user-based CF algorithm and the Euclidean Similarity metric with Neighborhood size equal to 50 and training *ratio* equal to 0.8 have the lowest RMSE and MAE, which means this technique returned the best predictions based on the selected data of our dataset. Therefore, we selected the *Euclidean similarity* metric as an appropriate technique for our user-based CF recommender engine.

Despite the fact that the obtained results are accurate, this technique was applied on a part of our dataset, which is the rating. However, our dataset contains more heterogeneous data such as students' interests, and demographic data. Thus, this approach is not adequate for the whole dataset. Therefore, in the next section, we implemented and evaluated the stand-alone DF technique, which is based on the user's demographic data.

6.2.2 The stand-alone DF recommender system

In this section, we experimented and evaluated the DF recommender system that is based on the demographic data. Here, the DF recommender system does not take into account the domain knowledge, user interests, and ratings in its recommendation process. Thus, the DF recommender system can provide recommendations before receiving any rating from the active students. Therefore, for many students generalizations with the demographic features seemed too general for the highly personalized recommendations. For example, not all 17-year-old male students who liked scientific courses at high school would prefer the same university major or career in the future. In addition, students with different opinions or an unusual interests result in low correlation coefficient with other students. Therefore, recommendations for that kind of students are very hard to generate. Thus, the recommendations that are based only on demographic data may lead to inaccurate predictions. This limitation is called the grey-sheep (de Campos et al., 2010) issue and is caused by odd recommendations since the student may have other features that do not match with any other student or community of students. An example of a grey-sheep issue is when a user neither agrees nor disagrees with any user or group of users. Grey-sheep issue can increase the error rate in recommendations and can affect the performance precision of the RS. Also, this issue possibly will negatively affect the predictions for the rest of the community in the dataset (Bruke, R., 2002). Besides, in some cases gathering of demographic data leads to privacy issues. However, in cooperation with social media websites, it becomes more effective as the private data is already published.

The experiments in this section were based only on students' demographic data such as student's language, gender and location, etc., showed inaccurate recommendations. Therefore, we considered the stand-alone DF approach unsuitable for our dataset, since it does not take into consideration the domain knowledge, students' interests, and rating history. In addition, the experiments revealed that no correlation found between the demographic data and the courses ratings in the university graduates' dataset.

In our case, demographic data is not enough on its own; it has to be combined with domain knowledge, courses' rating, and students' interests to generate more personalized recommendations. To overcome the problems and limitations of the stand-alone DF, different recommender systems approaches should be tested such as the KB and hybrid systems. Therefore, in the next section, we implemented and evaluated the CBR knowledge-based RS approach. This approach is based on the demographic data, domain knowledge, and students' interests.

6.2.3 The stand-alone KB recommender system supported by CBR

In section 6.2.1, we discussed how we used the graduates' rating dataset to recommend appropriate career domain that fit high school student interests. However, what if we need to recommend personalized recommendations to a high school student based on his/her interests and knowledge

and not on his/her coursers' ratings? Here comes the role of the KB recommender system that generates recommendations based on the domain knowledge instead of ratings.

In this section, we implemented the CBR knowledge-based approach to build the RS application and help high school students to cope their problems and needs. This CBR knowledge-based RS integrates interests, knowledge and demographic data to retrieve adequate recommendations.

Besides, using indexes to speed up the retrieval of data is a technique used in most database systems. Similarly, CBR uses indexes to speed up retrieval in case base. Case indexing involves assigning indexes to cases to facilitate their retrieval. The Indexes organize and label cases so that appropriate cases can be found when needed. The primary role of indexing is feature matching and retrieval of cases. Cases may be indexed by a prefixed or open vocabulary, and within a flat or hierarchical index structure. An index is composed of two terms; the index name and index value (Perner, 2019).

Different applications may have different case representational requirements and as the size of the case base increases, it becomes critical that the CBR system accesses the stored cases efficiently. To address such challenges, jColibri provides persistence mechanism through "*Connectors"* and in-memory organization for case base management. Different *Connectors* and data structures for in-memory organization are provided. jColibri separated the case storage from the indexing structure that reason with cases like retrieval or adaptation methods. That way, indexes can be built and methods can be configured without knowing how and where the cases are stored. *Connectors* are objects that know how to access and retrieve cases from the medium and return those cases to the CBR system in a uniform way. Currently, jColibri implements different *Connectors* to load a cases from data base, plain text file, XML file or Description Logics ontology. *Connectors* provide an abstraction mechanism that allows users to load cases from different storage sources in a transparent way. In-memory or indexing is the second layer of Case Base management. *In-memory* case organization is the data structure used to organize the cases once loaded into memory. Once cases are loaded they can be organized in several ways trying to improve the access to the case base: linear lists, trees, case retrieval nets, etc.

The following figure shows a high school student's query sample taken from the KB recommender system GUI.

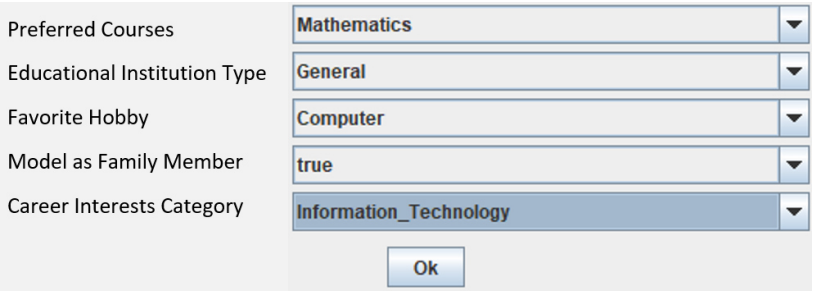


Figure 6.3: The CBR knowledge-based RS query sample

The following figure shows five recommendations suggested to a high school student based on his/her query. Each recommendation suggests a university/college, university major, and career domain. Thus, the high school student can select a suitable recommendation from the retrieved cases shown in the Figure 6.4 that match his/her interests.

Case ID	Prefered Course	Education	Hobby	Model as a Member of Family	Career Intereset	Academic Dicsipline	University	Occupation
622	Mathematics	General	Computer	TRUE	Information Technology	Computer Science	Lebanese Uni	Information Technology
676	Mathematics	General	Computer	TRUE	Information Technology	Computer Science	Lebanese Uni	Information Technology
56	Mathematics	General	Sports	TRUE	Information Technology	Computer Science	Lebanese Uni	Information Technology
66	Mathematics	General	Sports	TRUE	Information Technology	Computer Science	Lebanese Uni	Information Technology
125	Mathematics	General	Swimming	TRUE	Information Technology	CCE	AUL	Information Technology

Figure 6.4: The CBR knowledge-based RS retrieved solutions

To evaluate the accuracy of the recommendations we used the jColibri2 (Recio-García, J. A. et al., 2014) *NNScoringMethod* to measure the similarity rate. The following figure shows that the first two solutions are 100% similar to the above query and the other three solutions are 80% similar to the same query.

```
;y)][Solution: (null;Computer Science;Lebanese university;Information Technology)][Sol.Just.: null][Result: null] -> 1.0
;y)][Solution: (null;Computer science ;Lebanese university;Information Technology)][Sol.Just.: null][Result: null] -> 1.0
 [Solution: (null;Computer Science;Lebanese university;Information Technology)][Sol.Just.: null][Result: null] -> 0.8
 [Solution: (null;"Computer science;Math";Lebanese university)][Sol.Just.: null][Result: null] -> 0.8
;y)][Solution: (null;Computer communication engineering ;AUL;Information Technology)][Sol.Just.: null][Result: null] -> 0.8
```

Figure 6.5: The CBR knowledge-based RS recommendations' evaluations.

In order to improve further the recommendations' accuracy, we tried to integrate the ontology concepts similarity into the KB process. The next section shows how we implemented the CBR knowledge-based RS supported by the ontology.

6.2.4 The stand-alone KB recommender system supported by CBR and ontology

In this section, we experimented the advantages of KB recommender system that is supported by the ontology and CBR. In this approach, we used the career knowledge, higher education knowledge, students' interests and demographic data. Moreover, we integrated the domain ontology into the CBR knowledge-based RS to generate more personalized recommendations. The ontology in this approach encompasses the higher education, school, career and student profile concepts.

The experiments in this section are based on prior graduates' cases that are stored as instances in our ontology. The graduate cases were extracted from our survey based on many criteria such as using only data that are related to university graduates that have a university major related to their current job and their job meets their interests. The final refined case base encompasses 658 graduate cases.

This case base will be integrated into the ontology design and computed by the jColibri2 (Recio-García, J. A. et al., 2014) retrieval function. The main function of jColibri2 for computing the

retrieval of the most similar cases is the NNScoringMethod. This function performs a Nearest Neighbor numeric scoring comparing attributes. It uses global similarity functions such as the mean Average() to compare compound attributes and local similarity functions such as Detail() to compare simple attributes. For example, the Graduate case component is a compound attribute composed by several simple attributes (gender, language, hobby, country, etc.). When two cases are compared local computes the similarity between simple attributes and global computes some kind of average over the local similarities. Therefore, a global similarity function is assigned to the description like the average function. The NNScoringMethod will compute the similarity of each simple attribute and then compute the global similarity (the average of the simple similarities). The method returns a collection of RetrievalResult objects.

Most similar cases must be selected once they have been scored according to their similarity with the query. Usually, only the top k most similar cases are selected. This retrieval process that combines Nearest Neighbor scoring and top k selection is commonly called k-NN retrieval. Once the similarity function and weight are set for the attributes evaluateSimilarity() is executed obtaining a list of RetrievalResult objects that contain the most similar cases to the query. Finally, the most similar cases are obtained using the selectTopKRR().

In this system, we used the jColiri2 retrieval process to compare high school students' cases with university graduates' cases and find most similar cases in order to provide appropriate recommendations. The following figure shows a high school student query example. The query's attributes are selected from the instances that are saved in the ontology design.

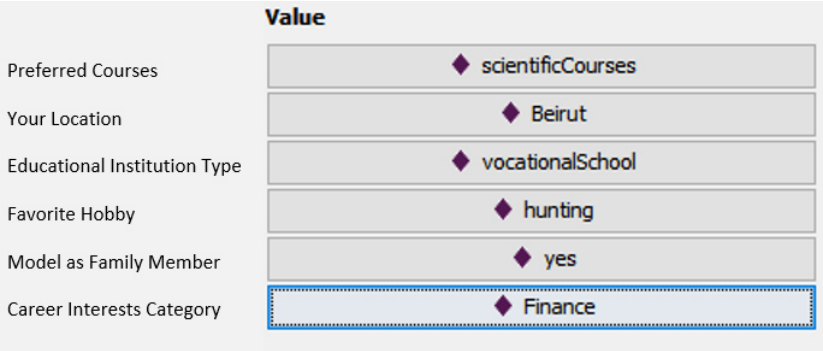


Figure 6.6: Active student query example

The scoring of the most similar cases in this process is computed based on the prior cases similarity with the query. At this point, the top k most similar cases will be retrieved. This retrieval process mixes the Nearest Neighbor scoring and top *k* selection techniques. Here, the calculation returns a value between (zero ~ one) showing the retrieved solution or case being less and most similar to the active query case. The following two figures show the most similar cases generated by the KB

recommender system based on the above student's query. The first retrieved case is 100% similar to the active user query case as shown in Figure 6.7.

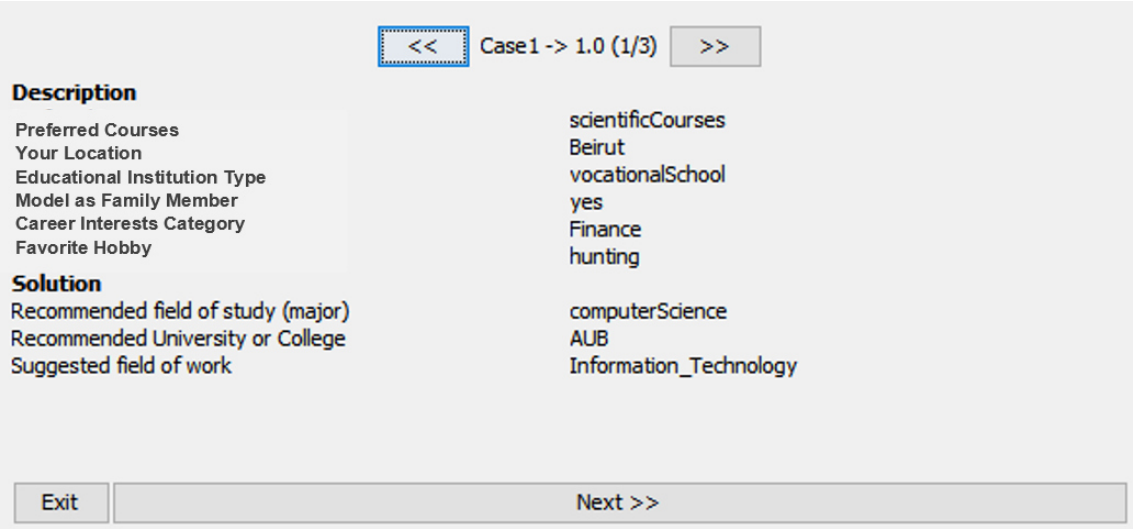


Figure 6.7: Most similar retrieved case to active student query

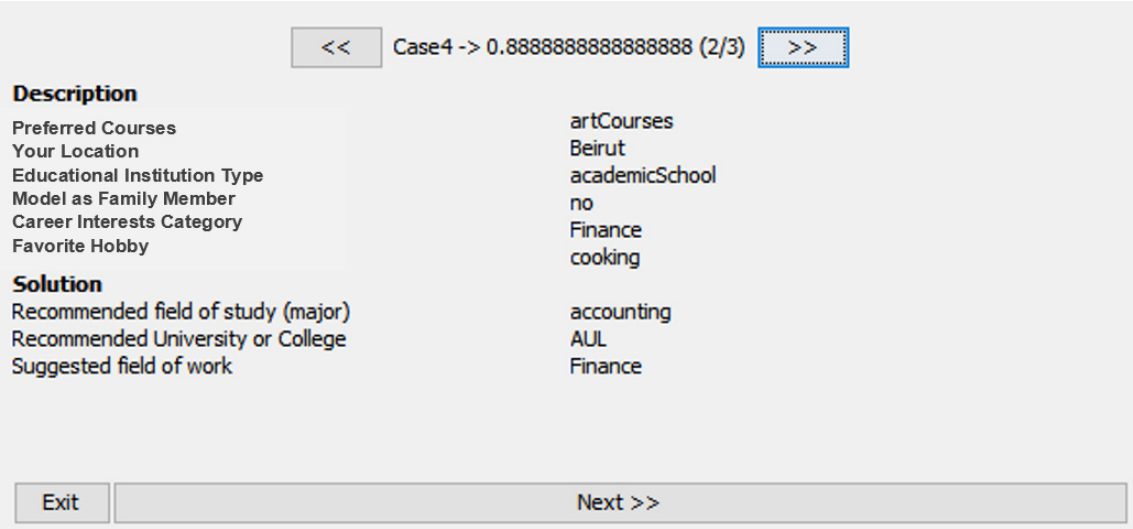


Figure 6.8: Second most similar case

Figure 6.8 shows the second most similar case to the active user query. This prior case is approximately 88% similar to the submitted query. This KB recommender system generated personalized recommendations to the high school student with the support of the ontology and CBR concepts. Hundreds of queries were tested and evaluated by this RS approach.

To evaluate the accuracy of this RS approach, the *HoldOutEvaluator* algorithm was implemented. The following two figures show the accuracy of this system based on the *HoldOutEvaluator* algorithm. More detail about this algorithm are presented in chapter 2 section 2.8.4.

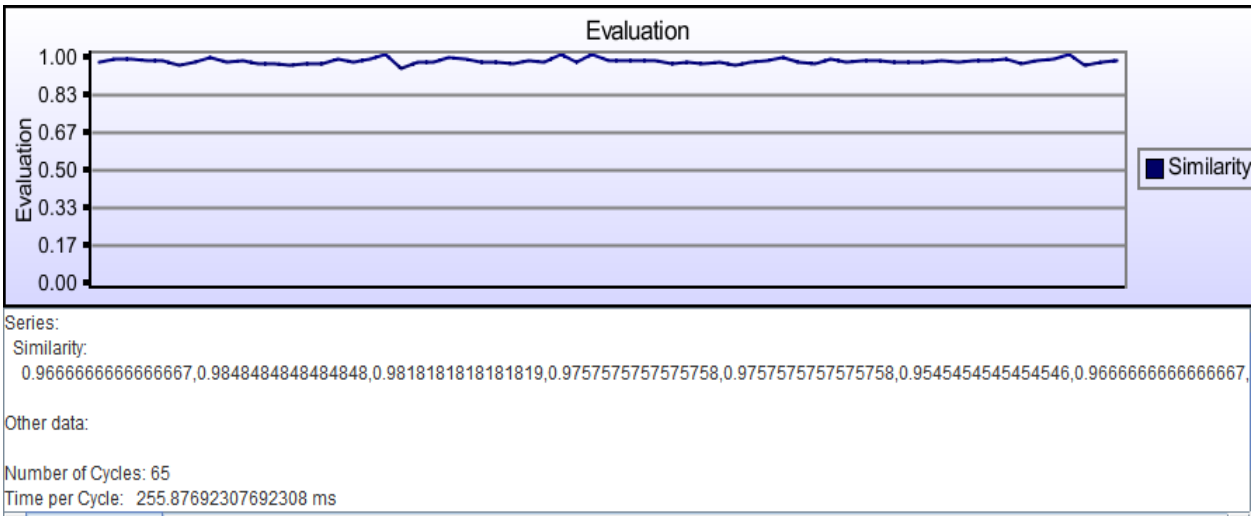


Figure 6.9: The evaluation result of our experiment approach number 4 (experiment 1)

The evaluation results in Figure 6.9 show the high accuracy of this system based on using 10 percent of the dataset for testing and performing the process several times through 65 cycles.

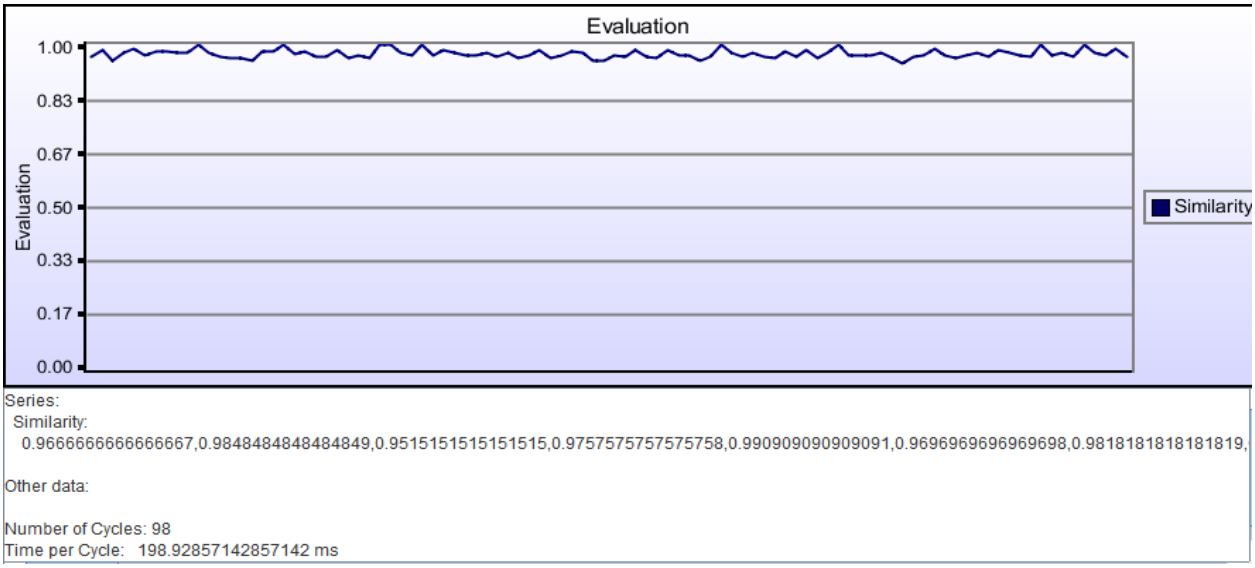


Figure 6.10: The evaluation result of our experiment approach number 4 (experiment 2)

The evaluation results in Figure 6.10 show the high accuracy of this system based on using 15 percent of the dataset for testing and performing the process several times through 98 cycles.

In addition, the following table shows the overall accuracy of this KB recommender system based on many criteria such as "*generating appropriate recommendations"* and "*retrieving the most similar cases*".

	System accuracy
Retrieving the most similar cases	98%
Generating appropriate top N recommendations based on the students' interest	94%

Table 6-2: The accuracy results of the system

The 98% accuracy of retrieving the most similar cases was calculated and generated based on many experiments. Also, the 94% accuracy of generating appropriate recommendations was calculated and generated based on many experiments.

Thus, this system is considered more efficient than previously tested recommender systems. In addition, the analysis and evaluations for this approach revealed that the ontology is very useful in supporting CBR knowledge-based RS. The ontology helped us to integrate the high school students interests, graduates knowledge, and high school and higher education knowledge into the KB recommender system, and conceptualize and implement it in a formal language. Likewise, the reuse of the ontology also benefits from its reliability and stability. Moreover, throughout the similarity calculation, the ontology permits to link the gap between the high school student's query and the case-based vocabulary.

In this section, we showed the effectiveness of the integration of the ontology into the CBR KB recommender system approach. This integration allowed the system to generate personalized recommendations. However, this approach doesn't take into consideration the students and graduates courses' ratings. Therefore, to benefit from the advantages of ontology, CBR, KB, and CF, we developed a uniform hybrid RS system that incorporates all these technologies. In the following section, this hybrid RS that is based on the KB and CF techniques is tested and evaluated.

6.2.5 The KB hybrid RS incorporated with the user-based CF and supported by CBR and ontology (COHRS)

In this section, the overall hybrid RS is tested. In addition, the mechanism of the connection between the CF and the KB techniques is described. Here, we used the methodology of combining the CF and KB techniques in order to recommend more personalized recommendations. This approach is a combination of the section 6.2.1 and 6.2.4 approaches. This hybridization aims to improve the recommendations and precision of the system. This combination allowed us to generate recommendation based on the domain knowledge, students' rating, interests, and demographic data.

The recommendation process starts by entering the high school courses' ratings of the high school student into the user-based CF system. Then, the CF engine will compare the graduates' ratings with the high school student ratings in order to find similarities and generates a career recommendaiton. As mentioned in section 6.2.1, the recommendation of our user-based CF is based on the *Euclidean distance* metirc. The following figure shows the CF graphical user interface

that is used to enter the courses' ratings. The high school student uses this system to rate his/her level on the school courses.

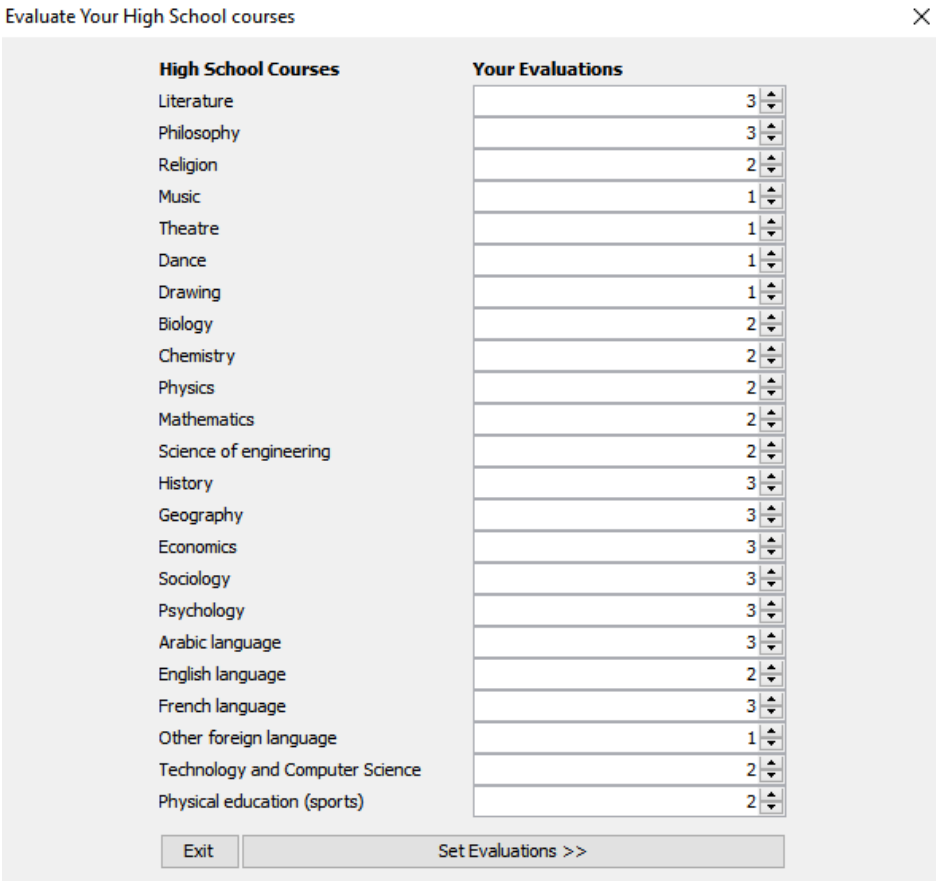


Figure 6.11: The CF graphical user interface

The CF recommedndation is integrated as a new feature into the KB system. This new feature is used in the KB system as a support knowledge to the high school student query. The role of the KB recommender systs is to generats the top N recommendations based on the high school student query and graduates prior cases that are saved as instances in the ontology. More details about COHRS architecture and mechanism are illustrated in Figure 5.1 in chapter 5. COHRS integrates a dataset that encompasses 658 graduate cases that represent only the university graduates that have a university major related to their current job and their job meets their interests. The following figure illustrates a query sample requested by a high school student in order to get recommendation toward the university paths.

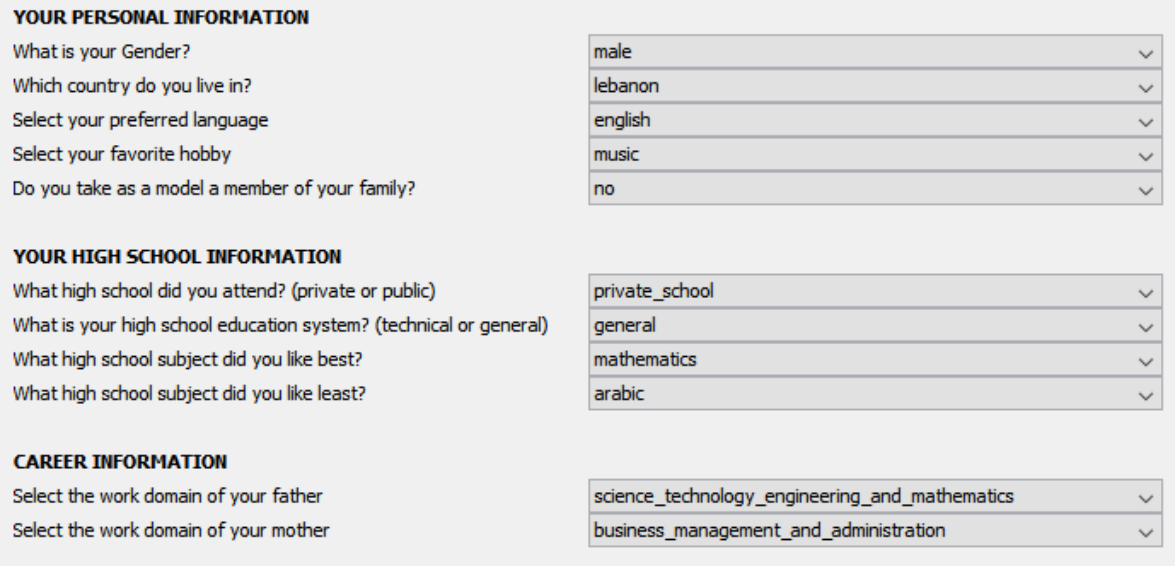


Figure 6.12: Query sample in our approach number 5

By implementing the ontology similarity and CBR retrieve method, this hybrid KB system can retrieve the most similar cases that fit the high school student interests. The following two figures show COHRS top N recommendations samples.

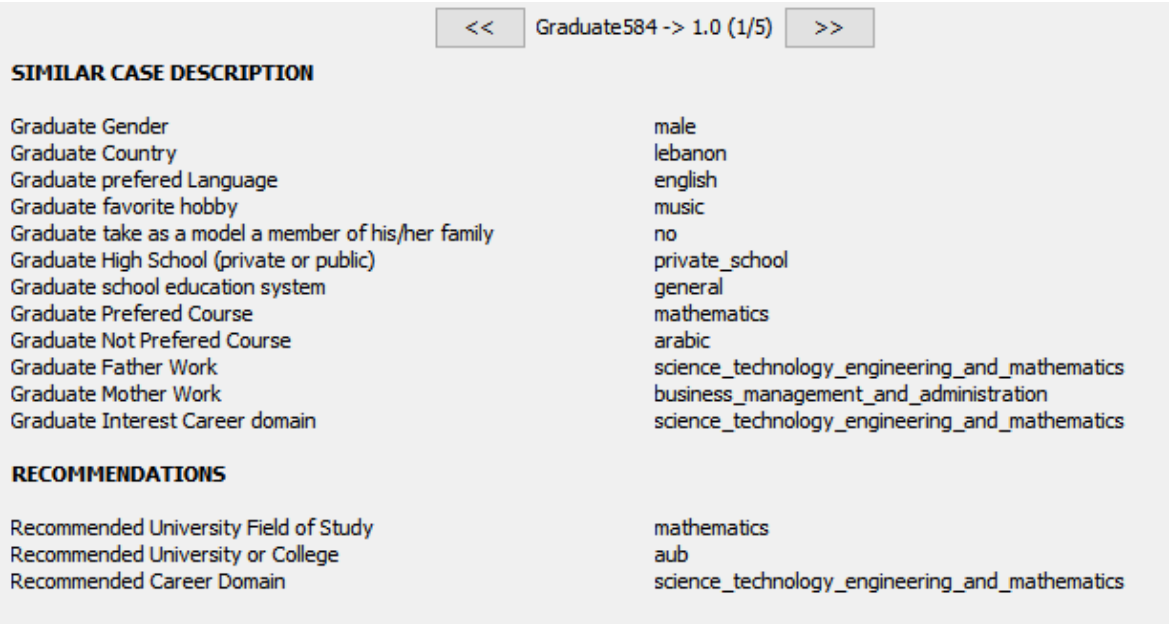


Figure 6.13: Most smilar case result in our approach number 5

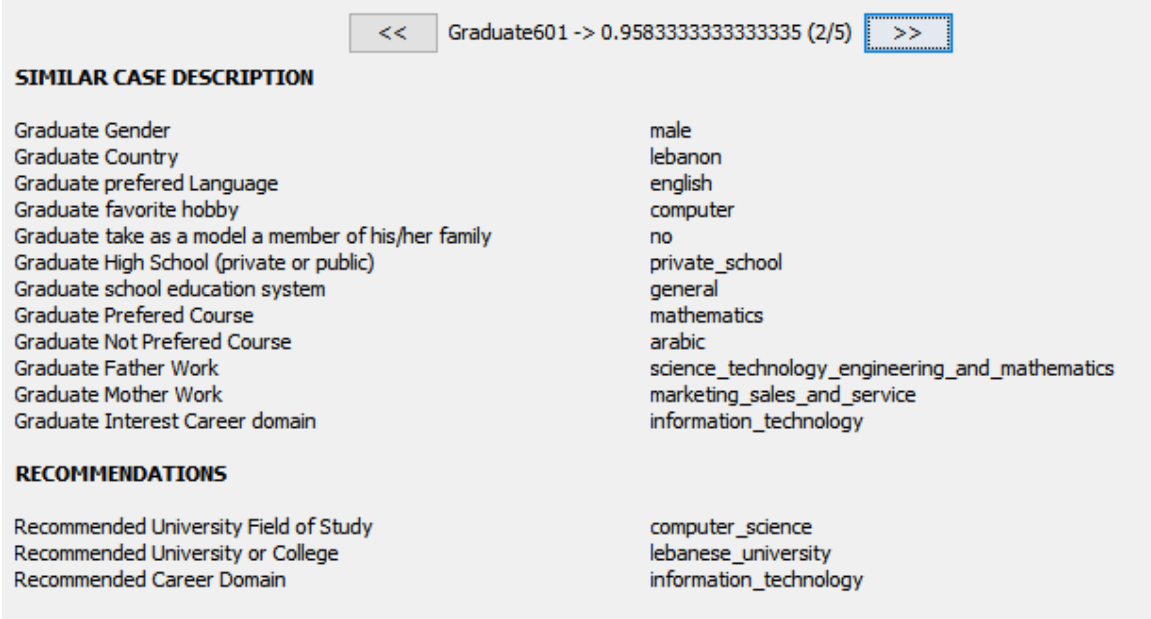


Figure 6.14: Second most smilar case result in our approach number 5

We explained in the precedent chapter that we used the Feature augmentation strategy which enable the recommender engine to incorporate two separate types of recommender algorithms in a way that the output of the first recommender is fed into the input of the second, this to improve the performance of the proposed hybrid system and made a significant contribution to the quality of recommendations. Figure 6.13 and Figure 6.14 show the "Graduate Interest Career Domain" feature that is integrated with the high school student query. This feature represents the CF recommendation that is generated based on the high school student and graduates' ratings. This feature is introduced in the KB recommender system.

Moreover, Figure 6.13 and Figure 6.14 show that graduate case 584 and graduate case 601 are the most similar cases to the currently high school student query. If we compare the active student query with the recommendations result, we notice that case 584 is more similar to the active student than case 601. This reveals that case 584 is totally similar to the active student query and case 601 is less similar with a difference in the hobby, "mother work" and "graduate interest career" attributes.

Furthermore, Figure 6.13 shows the recommendations of the KB hybrid RS as follows:

- University field of study= Mathematics
- University of college= AUB University
- Career domain= Science, Technology, Engineering, and Mathematics.

In addition, Figure 6.14 shows the recommendations of the KB hybrid RS as follows:

- University field of study= Computer Science
- University of college= Lebanese University
- Career domain= Information Technology

As mentioned before our proposed CBR and ontology based hybrid RS was developed and trained with 658 university graduate cases. Hundreds of queries were tested and evaluated by this RS approach. To evaluate the accuracy of COHRS approach, the *HoldOutEvaluator* and *SameSplitEvaluator* algorithms were implemented. More detail about these two algorithms are presented in chapter 2 section 2.8.4. The following two figures show the high precision of COHRS based on these two algorithms.

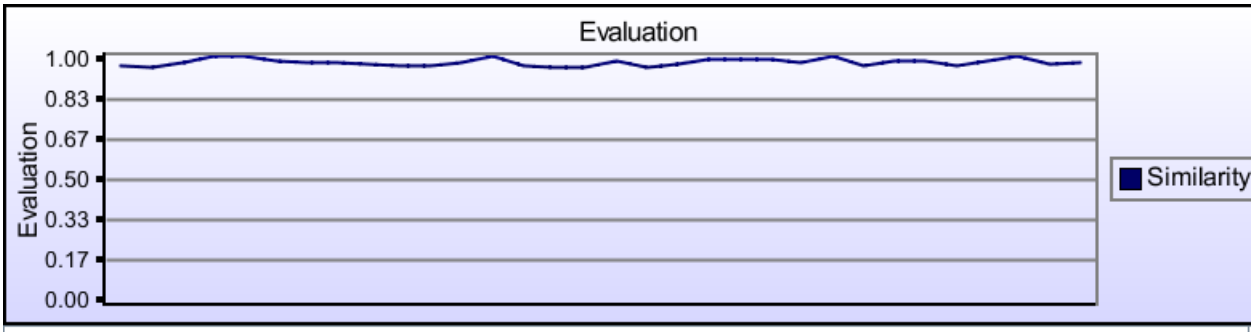


Figure 6.15: HoldOutEvaluator results

The evaluation results in Figure 6.15 show the high accuracy of this system based on using 5 percent of the dataset for testing and performing the process several times through 32 cycles.

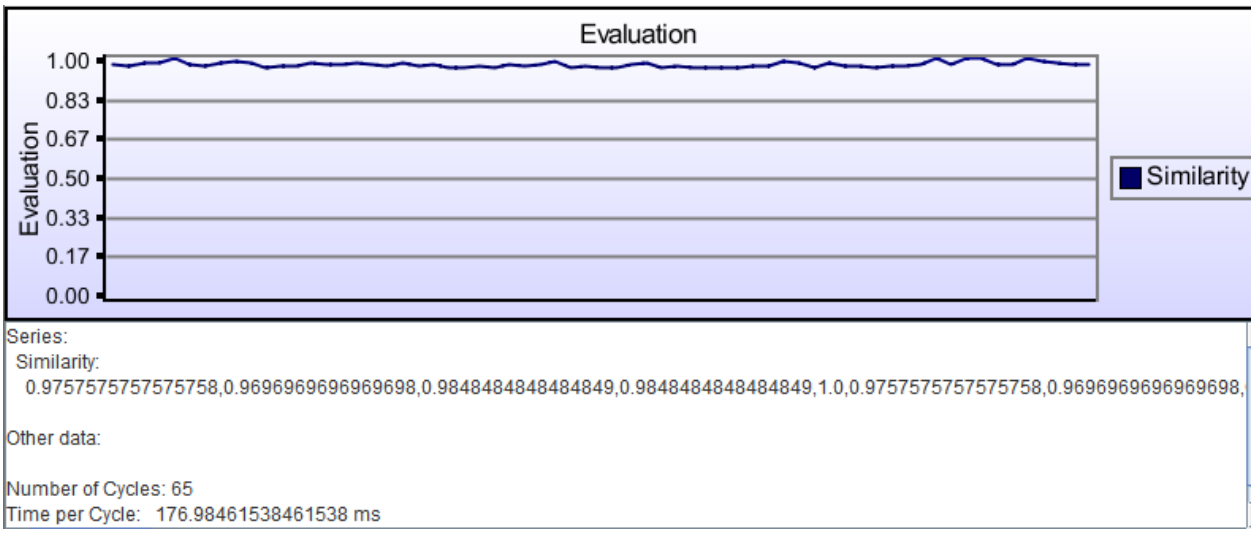


Figure 6.16: SameSplitEvaluator results

The evaluation results in Figure 6.16 show the high accuracy of this system based on using 10 percent of the dataset for testing and performing the process several times through 65 cycles.

Table 6-3 shows the high accuracy of COHRS based on two criteria namely the "*accuracy of retrieving the most similar cases*" and the "*accuracy of generating appropriate recommendations*".

	System accuracy
Retrieving the most similar cases	98%
Generating appropriate top N recommendations based on the students' interest	95%

Table 6-3: The accuracy results of COHRS

The 98% accuracy of "*retrieving the most similar cases*" was calculated and generated based on many experiments. Also, the 95% accuracy of "*generating appropriate recommendations"* was calculated and generated based on many experiments.

In addition, COHRS system was applied and tested on real-life cases. The test was conducted with a sample size of 60 high school students, 40 university students, and 40 university graduates. The objective of COHRS is to guide high school students toward university paths. Therefore, high school students were chosen to participate in the application experiments. Additionally, the university students were chosen since they have experienced the transition from school to university. Whereas, the university graduates were chosen as they already have passed this transition and know its outcomes.

The results are showed in Table 6-4 and Table 6-5. This test purpose is to find out whether the use of prior graduates knowledge can be applied to assist current high school students.

	Total number of participants	Not at all useful	Fairly useful	Very useful
High school students	60	3.333% (2 students)	3.333% (2 students)	93.333% (56 students)
University students	40	2.5% (1 university student)	5% (2 university students)	92.5% (37 university students)
University graduates	40	2.5% (1 university graduate)	2.5% (1 university graduate)	95% (38 university graduate)

Table 6-4: Results of COHRS interest

	Total number of participants	Dissatisfied	Satisfied	Very Satisfied
High school students	60	3.333% (2 students)	5% (3 students)	91.7% (55 students)
University students	40	5% (2 university students)	5% (2 university students)	90% (36 university students)
University graduates	40	5% (2 university graduates)	2.5 (1 university graduate)	92.5% (37 university graduates)

Table 6-5: Results of users' satisfaction

Table 6-4 shows that 93.6% of the users agreed that our hybrid RS is useful. The result was calculated as follows:

- Average = (93,333% + 92.5% + 95%) / 3 = 93.6%

Table 6-5 shows that 91.4% of the users were very satisfied with the RS recommendations. The result was calculated as follows:

- Average = (91,7% + 90% + 92.5%) / 3 = 91.4%

All our experiments showed the efficiency of our user-based CF system in supporting the KB recommender system. In addition, our analysis revealed that the hybridization approach of COHRS is the most adequate solution to our high-dimensional dataset that encompasses more than 50 heterogeneous attributes. Furthermore, we noticed that the more knowledge we acquire the more effective the ontology-based hybrid RS could be. Finally, we deduced that our COHRS approach is an effective mechanism for designing KB hybrid RS with high dimensional datasets.

6.3 Synthesis

The following table shows the comparative analysis of the five tested approaches presented in this chapter.

RS technique	***Data type***	***Advantages***	***Disadvantages***	***Effectiveness***
Standalone user-based or item-based CF	Users' ratings	No need to domain knowledge. It captures the change in user interests and preferences over time.	Has many limitations such as cold-start problem, data sparsity and grey-sheep…	This technique is not adequate to our dataset since it computes only rating history and has many limitations. Additionally, it recommends only career domains.
Standalone DF	Demographic data	No need to domain knowledge and history of user ratings. It reduces the new user problem of the CF.	Lack of demographic data and users are reluctant to disclose it. In addition, demographic data in combination with item ratings are difficult to acquire. Gathering of demographic data leads to privacy issues. It has Grey-sheep issue.	This technique is not adequate to our dataset since it integrates only demographic data. In addition, our analysis revealed the DF weakness in generating personalized recommendations.
Standalone KB supported by CBR.	Domain Knowledge of prior cases	It is a valuable tool when the item is infrequently used. It is not dependent on the ratings.	It is hard to acquire domain knowledge.	This technique is adequate to our dataset and provided correct returns. However, it needs further improvement in order to retrieve more accurate similar cases. Therefore, we worked on incorporating this approach with the ontology concepts similarity.
Standalone KB supported by CBR and ontology.	Domain Knowledge of prior cases + ontology	It is a valuable tool when the item is infrequently used. It is not dependent on the ratings.	It is hard to acquire domain knowledge. It requires knowledge engineering skills.	This technique is adequate to our dataset and provided accurate returns. Moreover, it generated better recommendations than the other systems. However, this approach does not compute ratings. Therefore, it should be

				incorporated with the CF technique in a hybrid RS in order to compute knowledge and ratings.
KB + user-based CF supported by CBR and ontology. (COHRS)	Users' ratings + Domain Knowledge of prior cases + demographic data + ontology	It is useful in combination with other forms of recommender systems. It is a valuable tool when the item is infrequently used. It computes the semantic similarity between the ontology concepts. Knowledge is represented in the form of ontology. Moreover, recommendations are generated from two RS techniques. Furthermore, can treat high dimensional datasets.	It is hard to acquire domain knowledge. It requires knowledge engineering skills.	This hybridization technique is the most suitable recommendation technique for our dataset since it computes the returns based on the domain knowledge, high school student's profile, ratings, interests, and demographic data. Also, it provides high accuracy recommendations since it integrates 4 core techniques namely the (KB, CF, CBR, and ontology). The generated recommendations are a university, university major, and career domain. Furthermore, it shows high precision in treating heterogeneous data types and high dimensional datasets.

Table 6-6: Comparative analysis of recommender systems

6.4 Conclusion

In this chapter, we implemented and evaluated four RS techniques namely the (CF, DF, KB, and KB Hybrid RS) in order to demonstrate the efficiency of COHRS. Thus, we experimented and evaluated five recommendation approaches namely the stand-alone user-based and item-based CF techniques, stand-alone DF technique, stand-alone KB technique supported by CBR, stand-alone KB technique supported by CBR and ontology, and KB Hybrid RS incorporated with the user-based CF technique and supported by the CBR and ontology.

In conclusion, our experiments show the efficiency of COHRS in recommending accurate recommendations. In addition, the analysis reveals that COHRS is an adequate approach for achieving our objectives since it has the capability to process the heterogeneity of our data. Moreover, we deduced that this hybrid system is a promising tool for guiding high school students toward the university paths.

The novelty of our approach focuses precisely on CBR and ontology based hybrid RS within the higher education domain, of which to the best of our information, no research has been conducted to present this problematic. Finally, we considered COHRS approach an efficient mechanism for

designing KB hybrid recommender systems since it generates precise and personalized recommendations.

PART IV

CONCLUSION, OUR PERSPECTIVES, AND FUTURE RESEARCH

In this section, the summary of this research's achievements, our perspectives and future research are presented.

Conclusion

The orientation programs in most schools are not well designed to cater to students' varied needs. In addition to that, the complexity of life and the instability of the job market strongly affects youth when choosing a university major. Faced with these problems, high school students feel lost when choosing their majors at the university. Besides, the explosive evolution of knowledge and data on the Web network with the growth of innovative electronic machines has made the WWW information increasingly significant in most internet users' life. As a result, internet users are forced to take inappropriate decisions when searching the Web due to an incapability to deal with the massive volumes of data.

Thus, our study focuses on developing a novel hybrid RS named COHRS (CBR and Ontology based Hybrid Recommender System) that enables students to explore top N recommendations based on their fields of interest. The system general objective is to assist learners in making the right decision when selecting their university/college, university major, and career choices. The major purposes of our research are to recommend accurate and personalized recommendations to high school students based on their (interests, demographic data, courses' ratings, and domain knowledge); address the limitations of the basic RS filtering techniques; and develop a novel hybrid RS approach that incorporates four core technologies namely the *KB, CF, CBR,* and ontology. In order to achieve the desired purposes, this research has answered the following questions:

- How can we generate personalized recommendations to high school students toward higher education choices?
- How can we incorporate the KB and CF recommendation engines to collaborate in a uniform hybrid RS?
- How can we integrate the ontology into the KB recommender system to improve the recommendation and enhance the objects matching in the KB recommendation process?
- How can we integrate the CBR system into the KB recommender system and interconnect it with the domain ontology to increase the accuracy of recommendations?
- How can we treat and integrate heterogeneous data types into the hybrid recommendation process?
- How can we overcome the limitations of the traditional recommender system techniques?
- How can we treat high dimensional datasets via a hybrid RS engine?

This work involves many phases described as follows:

(1) Acquiring the required data from an online survey that is used as resources to our analysis process. This survey includes more than 50 attributes that involves nominal, ordinal, and numerical data types. The answers to the survey questions encompasses heterogeneous data such as graduates' demographic data, graduates interests, graduates courses' rating and domain knowledge.
(2) Working on cleaning, transforming, and refining the acquired data using many data-preprocessing techniques.
(3) Clustering graduates' trajectories using three different clustering techniques namely the *FCA, K-modes,* and *Hierarchical.*
(4) Experimenting and analyzing many different recommender-filtering techniques such as KB, KB with CBR, KB with CBR and ontology, DF, user-based CF, item-based CF, and hybrid RS.
(5) Proposing and developing a KB hybrid RS named COHRS supported by CBR and ontology. This hybrid RS makes use of high school students and graduates courses' ratings, demographic data, higher education and career knowledge in the recommendation process in order to recommend personalized recommendations.

Our research has made important contributions in the field of hybrid recommender systems, as detailed in the following:

COHRS incorporates the user-based CF and KB techniques in a uniform system supported by the ontology and CBR technologies. These two recommendation-filtering techniques interconnect through the "Feature Augmentation" hybridization strategy to recommend personalized recommendations. This hybrid system integrates the domain knowledge into its KB recommender system that generates recommendations based on the ontology similarity and CBR case matching. The ontological knowledge represents the high school students' profiles, higher education, and career domains. The higher education ontology represents the knowledge about the university/college, university majors, etc. Whereas the career ontology represents the career domain and the user's profile ontology describes the user's model and interests. Besides, the user-based CF system uses students and graduates' ratings to generate recommendations based on the Euclidean Distance similarity metric.

Additionally, COHRS has the capability to integrate and compute heterogeneous data types. It can treats nominal, ordinal, rating, and numerical data types to generate precise recommendations.

The experimental studies show that COHRS has reduced the limitations of traditional RS filtering techniques. Additionally, COHRS achieved higher performance than other recommender system approaches. Thanks to the CBR and ontology that increased the recommendations' accuracy. As stated in chapter 6, this novel approach outperforms other recommendation filtering techniques by generating accurate and personalized recommendations.

In conclusion, this research has reached its objectives. The analysis revealed that the KB technique incorporation with the user-based CF technique is an adequate approach to process multi-data-types datasets. Furthermore, we deduced that this hybrid system is a promising tool for guiding high school students toward higher education paths.

The novelty of our method focuses precisely on CBR and ontology based hybrid RS within the higher education domain, of which to the best of our information, no research has been conducted to present this problematic. Finally, we considered COHRS approach an efficient mechanism for designing KB hybrid recommender systems since it generates precise and personalized recommendations.

Our perspectives and future research

Recommender systems should be always developed and improved due to the extensive growth of the *WWW* specifically in e-commerce, e-learning, medicine, etc. Existing recommender systems that use domain knowledge instead of users' ratings need more enhancement specially that are based on multi-data-type datasets.

The key features of our developed hybrid RS do not scope the following aspects: The *Natural Language Processing (NLP) and Ontology Learning*.

Thus, in future work the following directions can be taken into consideration:

Suggest how *NLP* methods could be employed for the improvement of COHRS. NLP has the capability to understand natural language. Although we used the WordNet lexical database to search for synonyms and correct the misspelling of data entry by students, this experiment taken into account only the synonyms of words. Several *NLP* methods (Alharthi and Inkpen, 2019), such as syntactic and stylometric features, word embedding, and extracting lexical could be used in RS filtering techniques for the recommendation of items. *NLP* techniques could be integrated into COHRS to identify university majors or courses dealing with the same interests and determines the semantic emotion of the course's reviews. In addition, NLP could serves a student query with summarized descriptions from all sources. Furthermore, NLP could be a component of a RS for feature extraction using TF-IDF and text classification.

Study *Ontology Learning* (Wong et al., 2012) or *Ontology Acquisition* that is the automatic or semi-automatic construction of ontology, comprising acquiring the corresponding expressions/terms of any domain and the relations between the concepts that these

terms describe from a collection of natural language text, and embedding them with an ontology representation for facile retrieval and reuse.

Additionally, apply different hybridization strategies (Bruke, R., 2002) such as *Weighted, Switching, Mixed, Cascade, etc.* These strategies can help in changing the RS mechanism that affect the work of COHRS recommender engine. In addition, the integration of new different filtering techniques with COHRS such as CB, model-based CF, etc. can improve the recommendation returns. For example, by incorporating the CB technique with COHRS mechanism and integrating the content of the available universities' descriptions, university majors' descriptions, courses' descriptions, and careers' descriptions, the recommendations accuracy could be more precise and personalized. Here, the content could also necessitate NLP to make use of syntactic and semantic characteristics.

Thus, COHRS can be extended by incorporating the *NLP* and automatic ontology construction technique to make recommendations more effective. Along these directions, COHRS key features should simplify the hard work of the ontology construction; and reduce the limitations of typical recommender systems through the integration of the *NLP* technology and incorporation of different filtering techniques such as the CB. Finally, to keep COHRS's data resources updated, we will ask users to provide us periodically with their demographic data, interests, preferences, and actual career situations

Publications related to the thesis

Paper

Charbel Obeid, Inaya Lahoud, Hicham El Khoury and Pierre-Antoine Champin. 2018. *Ontology-based recommender system in higher education.* The Web Conference Companion (WWW 2018), April 23-27, 2018, Lyon, France, ACM, New York, NY, 4 pages. DOI: https://doi.org/10.1145/3178876.3191533.

Journals

Journal 1: Charbel Obeid, Christine Lahoud, Hicham El Khoury, Pierre-Antoine Champin. *Conceptual Clustering of University Graduate Students' Trajectories Using Formal Concept Analysis: A Case Study in Lebanon*. International Journal of Continuing Engineering Education and Life-Long Learning, Inderscience, 2020, 30 (1), pp.1. ⟨10.1504/IJCEELL.2020.10027147⟩. ⟨hal-02737553⟩

Journal 2: Charbel Obeid, Christine Lahoud, Hicham El Khoury, Pierre-Antoine Champin. *A hybrid recommender system approach for students' academic advising supported by case-based reasoning and ontology*. Computer Science and Information Systems Journal, 2022.

References

Agarwal, G., Bahuguna, D.H., Agarwal, D.A., 2017. Solving Cold-Start Problem in Recommender System Using User Demographic Attributes. International Journal on Emerging Technologies (Special Issue NCETST-2017) 8(1): 55-61 7.

Aggarwal, C.C., 2016a. Recommender Systems: The Textbook. Springer International Publishing. https://doi.org/10.1007/978-3-319-29659-3

Aggarwal, C.C., 2016b. Content-Based Recommender Systems, in: Aggarwal, C.C. (Ed.), Recommender Systems: The Textbook. Springer International Publishing, Cham, pp. 139–166. https://doi.org/10.1007/978-3-319-29659-3_4

Alam, M., Le, T.N.N., Napoli, A., 2016. LatViz: A New Practical Tool for Performing Interactive Exploration over Concept Lattices 12.

Alharthi, H., Inkpen, D., 2019. Study of linguistic features incorporated in a literary book recommender system, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC '19. Association for Computing Machinery, Limassol, Cyprus, pp. 1027–1034. https://doi.org/10.1145/3297280.3297382

Allani, S., Yeferny, T., Chbeir, R., 2018. A scalable data dissemination protocol based on vehicles trajectories analysis. Ad Hoc Networks 71, 31–44. https://doi.org/10.1016/j.adhoc.2017.12.003

Andreasik, J., 2017. The Architecture of the Intelligent Case-Based Reasoning Recommender System (CBR RS) Recommending Preventive/Corrective Procedures in the Occupational Health and Safety Management System in an Enterprise 16.

Bagchi, S., 2015. Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout. Procedia Computer Science, Big Data, Cloud and Computing Challenges 50, 229–234. https://doi.org/10.1016/j.procs.2015.04.055

Bahramian, Z., Abbaspour, R.A., 2015. AN ONTOLOGY-BASED TOURISM RECOMMENDER SYSTEM BASED ON SPREADING ACTIVATION MODEL. https://doi.org/10.5194/isprsarchives-XL-1-W5-83-2015

Bal, M., Bal, Y., Ustundag, A., 2011. Knowledge Representation and Discovery Using Formal Concept Analysis: An HRM Application 6.

Bartha, P., 2019. Analogy and Analogical Reasoning, in: Zalta, E.N. (Ed.), The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University.

Bendakir, N., Aïmeur, E., 2006. Using Association Rules for Course Recommendation 10.

Benouaret, I., 2017. Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels.

Bilgin, G., Dikmen, I., Birgonul, M.T., 2014. Ontology Evaluation: An Example of Delay Analysis. Procedia Engineering, Selected papers from Creative Construction Conference 2014 85, 61–68. https://doi.org/10.1016/j.proeng.2014.10.529

Billsus, D., Pazzani, M.J., 2000. User Modeling for Adaptive News Access. User-Modeling and User-Adapted Interaction 147-180.

Blanco-Fernández, Y., López-Nores, M., Gil-Solla, A., Ramos-Cabrer, M., Pazos-Arias, J.J., 2011. Exploring synergies between content-based filtering and Spreading Activation techniques in knowledge-based recommender systems. Information Sciences 181, 4823–4846. https://doi.org/10.1016/j.ins.2011.06.016

Bobadilla, J., Ortega, F., Hernando, A., Bernal, J., 2012. A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems 26, 225-238.

Bousbahi, F., Chorfi, H., 2015. MOOC-Rec: A Case Based Recommender System for MOOCs. Procedia - Social and Behavioral Sciences 195, 1813–1822. https://doi.org/10.1016/j.sbspro.2015.06.395

Boyce, S., Pahl, C., 2007. Developing Domain Ontologies for Course Content 14.

Breese, J.S., Heckerman, D., Kadie, C., 2013. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. arXiv:1301.7363 [cs].

Bridge, D., Ferguson, A., 2002. Diverse Product Recommendations Using an Expressive Language for Case Retrieval, in: Craw, S., Preece, A. (Eds.), Advances in Case-Based Reasoning, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 43–57. https://doi.org/10.1007/3-540-46119-1_5

Bruke, R., 2002. Hybrid Recommender Systems. Survey and Experiments. User Modeling and User-Adapted Interaction 4:331-370.

Bürger, T., Simperl, E., 2008. Measuring the Benefits of Ontologies, in: Meersman, R., Tari, Z., Herrero, P. (Eds.), On the Move to Meaningful Internet Systems: OTM 2008 Workshops, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 584–594. https://doi.org/10.1007/978-3-540-88875-8_82

Burke, R., 2007. Hybrid Web Recommender Systems, in: Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 377–408. https://doi.org/10.1007/978-3-540-72079-9_12

Burke, R., 2000. Knowledge-based recommender systems. Encyclopedia of Library and Information Science, volume 69 23.

Burke, R., 1999. Integrating Knowledge-Based and Collaborative-Filtering Recommender Systems. In Proceedings of the Workshop on AI and Electronic Commerce, pp. 69–72 4.

Burke, R.D., Hammond, K.J., Yound, B.C., 1997. The FindMe approach to assisted browsing. IEEE Expert 12, 32–40. https://doi.org/10.1109/64.608186

Capuano, N., Gaeta, M., Ritrovato, P., Salerno, S., 2014. Elicitation of latent learning needs through learning goals recommendation. Computers in Human Behavior 30, 663–673. https://doi.org/10.1016/j.chb.2013.07.036

Chavarriaga, O., Florian-Gaviria, B., Solarte, O., 2014. A Recommender System for Students Based on Social Knowledge and Assessment Data of Competences, in: Rensing, C., de Freitas, S., Ley, T., Muñoz-Merino, P.J. (Eds.), Open Learning and Teaching in Educational Communities, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 56–69. https://doi.org/10.1007/978-3-319-11200-8_5

Cheng, W., Yin, G., Dong, Y., Dong, H., Zhang, W., 2016. Collaborative Filtering Recommendation on Users' Interest Sequences. PLoS One 11. https://doi.org/10.1371/journal.pone.0155739

Christodoulou, P., Christodoulou, K., Andreou, A.S., 2017. A real-Time targeted recommender system for supermarkets.

Christodoulou, P., Lestas, M., Andreou, A.S., 2013. A Dynamic Web Recommender System Using Hard and Fuzzy K-Modes Clustering, in: Papadopoulos, H., Andreou, A.S., Iliadis, L., Maglogiannis, I. (Eds.), Artificial Intelligence Applications and Innovations, IFIP Advances in Information and Communication Technology. Springer, Berlin, Heidelberg, pp. 40–51. https://doi.org/10.1007/978-3-642-41142-7_5

Cobos, C., Rodriguez, O., Rivera, J., Betancourt, J., Mendoza, M., León, E., Herrera-Viedma, E., 2013. A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes. Information Processing & Management, Personalization and Recommendation in Information Access 49, 607–625. https://doi.org/10.1016/j.ipm.2012.12.002

Conceptual Design of Document NoSQL Database with Formal Concept Analysis, 2016. . APH 13. https://doi.org/10.12700/APH.13.2.2016.2.13

Cunningham, P., Bergmann, R., Schmitt, S., Traphöner, R., Breen, S., Smyth, B., 2001. WEBSELL: Intelligent Sales Assistants for the World Wide Web 6.

Das, R., Zaheer, M., Thai, D., Godbole, A., Perez, E., Lee, J.-Y., Tan, L., Polymenakos, L., McCallum, A., 2021. Case-based Reasoning for Natural Language Queries over Knowledge Bases. arXiv:2104.08762 [cs].

de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Rueda-Morales, M.A., 2010. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning 51, 785–799. https://doi.org/10.1016/j.ijar.2010.04.001

Do, P., 2010. Model-based Approach for Collaborative Filtering 13.

Domingue, J., Fensel, D., Hendler, J.A, 2011. Introduction to the Semantic Web Technologies. Handbook of Semantic Web Technologies, Springer-Verlag Berlin Heidelberg, pp. 3-41.

Dubois, D., Hullermeier, E., Prade, H., 2002. Fuzzy set-based methods in instance-based reasoning. IEEE Transactions on Fuzzy Systems 10, 322–332. https://doi.org/10.1109/TFUZZ.2002.1006435

Duque Méndez N.D., Rodríguez Marín P.A., Ovalle Carranza D.A., 2018. Intelligent Personal Assistant for Educational Material Recommendation Based on CBR [WWW Document]. . Intelligent Systems Reference Library, vol 132. Springer, Cham.

Egho, E., Jay, N., Raïssi, C., Napoli, A., 2011. A FCA-based analysis of sequential care trajectories. Presented at the The Eighth International Conference on Concept Lattices and their Applications - CLA 2011.

Esfahani, M.H., Alhan, F.K., 2013. New hybrid recommendation system based On C-Means clustering method, in: The 5th Conference on Information and Knowledge Technology. Presented at the The 5th Conference on Information and Knowledge Technology, pp. 145–149. https://doi.org/10.1109/IKT.2013.6620054

Farzan, R., Brusilovsky, P., 2006. Social navigation support in a course recommendation system, in: Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'06. Springer-Verlag, Dublin, Ireland, pp. 91–100. https://doi.org/10.1007/11768012_11

Fellbaum, Christiane, 2005. WordNet and wordnets. In: Brown, Keith et al. (eds.). Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670.

Fennell, Philip., 2013. Extremes of XML, XML London.

Fesenmaier, D.R., Ricci, F., Erwin, S., Karl, W., Cristiano, Z., 2003. DIETORECS: Travel Advisory for Multiple Decision Styles, in: Frew, A.J., Hitz, M., O'Connor, P. (Eds.), Information and Communication Technologies in Tourism 2003. Springer Vienna, Vienna, pp. 232–241. https://doi.org/10.1007/978-3-7091-6027-5_25

Frauenfelder, M., 2004. Sir Tim Berners-Lee. Technology Review, 107(8), 40.

G. Adomavicius, N. Manouselis, Y. Kwon, 2011. Multi-criteria recommender systems in Recommender Systems Handbook. ed. by F. Ricci, L.Rokach,B. Shapira (eds.) (SpringerUS).

Ghazarian, S., Nematbakhsh, M.A., 2015. Enhancing memory-based collaborative filtering for group recommender systems. Expert Systems with Applications 42, 3801–3812. https://doi.org/10.1016/j.eswa.2014.11.042

Giacomelli, P., 2013. Apache Mahout Cookbook. Packt Publishing, Birmingham.

Gil, A., Rodríguez, S., De la Prieta, F., De Paz, J.F., Martín, B., 2012. CBR Proposal for Personalizing Educational Content, in: Vittorini, P., Gennari, R., Marenzi, I., de la Prieta, F., Rodríguez, J.M.C. (Eds.), International Workshop on Evidence-Based Technology Enhanced Learning, Advances in Intelligent and Soft Computing. Springer, Berlin, Heidelberg, pp. 115–123. https://doi.org/10.1007/978-3-642-28801-2_14

Göker, M.H., Thompson, C.A., 2000. The Adaptive Place Advisor: A Conversational Recommendation System.

Gomez-Albarran, M., Jimenez-Diaz, G., 2009. Recommendation and Studentsâ?? Authoring in Repositories of Learning Objects: A Case-Based Reasoning Approach. International Journal of Emerging Technologies in Learning (iJET) 4. https://doi.org/10.3991/ijet.v4s1.797

Gómez-Pérez, A., Fernandez-Lopez, M., Corcho, O., 2004. Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition, Advanced Information and Knowledge Processing. Springer-Verlag, London. https://doi.org/10.1007/b97353

Gorshkov, S., 2015. Building Ontologies for Agent-Based Simulation, in: Tan, Y., Shi, Y., Buarque, F., Gelbukh, A., Das, S., Engelbrecht, A. (Eds.), Advances in Swarm and Computational Intelligence, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 185–193. https://doi.org/10.1007/978-3-319-20469-7_21

Goštautaitė, D., Kurilov, J., 2021. Comparative Analysis of Exemplar-Based Approaches for Students' Learning Style Diagnosis Purposes. Applied Sciences 11, 7083. https://doi.org/10.3390/app11157083

Goux, D. et al., 2017. Adjusting Your Dreams? High School Plans and Dropout Behavior. The Economic Journal. https://doi.org/10.1111/ecoj.12317

Gower, J.C., 1971. A General Coefficient of Similarity and Some of Its Properties. Biometrics 27, 857–871. https://doi.org/10.2307/2528823

Greenacre, M.J., 1984. Theory and applications of correspondence analysis.

Grimm, S., Abecker, A., Völker, J., Studer, R., 2011. Ontologies and the Semantic Web, in: Domingue, J., Fensel, D., Hendler, J.A. (Eds.), Handbook of Semantic Web Technologies. Springer, Berlin, Heidelberg, pp. 507–579. https://doi.org/10.1007/978-3-540-92913-0_13

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220. https://doi.org/10.1006/knac.1993.1008

Gündüz-Ögüdücü, Ş., 2010. Web Page Recommendation Models: Theory and Algorithms. Synthesis Lectures on Data Management 2, 1–85. https://doi.org/10.2200/S00305ED1V01Y201010DTM010

Haarslev, V., M˙oller, R., 2001. RACER system description. In Proc. of the Int. JointConf. on Automated Reasoning (IJCAR-01). LNCS, Springer.

Hailemariam, L., Zhao, C., Joglekar, G., Whittinghill, D., Jain, A., Venkatasubramanian, V., Reklaitis, G.V., Morris, K.R., Basu, P.K., 2006. An Ontology-Based Information Management System for Pharmaceutical Product Development 11.

Hans, S., 2016. A Novel Evaluation Scheme using Formal Concept Analysis [WWW Document]. URL /paper/A-Novel-Evaluation-Scheme-using-Formal-Concept-Hans/5f1be7fe38d91d8d4fdec9a9735c2893b9c1524d (accessed 6.29.20).

Hao, J., Bouzouane, A., Gaboury, S., 2018a. Recognizing Multi-Resident Activities in Non-intrusive Sensor-Based Smart Homes by Formal Concept Analysis. Neurocomputing. https://doi.org/10.1016/j.neucom.2018.08.033

Hao, J., Bouzouane, A., Gaboury, S., 2018b. Recognizing multi-resident activities in non-intrusive sensor-based smart homes by formal concept analysis. Neurocomputing 318, 75–89. https://doi.org/10.1016/j.neucom.2018.08.033

Haruechaiyasak, C., Tipnoe, C., Kongyoung, S., Damrongrat, C., Angkawattanawit, N., 2005. A Dynamic Framework for Maintaining Customer Profiles in E-Commerce Recommender Systems, in: 2005 IEEE International Conference on E-Technology, e-Commerce and e-Service. Presented at the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, IEEE, Hong Kong, China, pp. 768–771. https://doi.org/10.1109/EEE.2005.8

Hassel, S., Ridout, N., 2018. An Investigation of First-Year Students' and Lecturers' Expectations of University Education. Frontiers in Psychology 8, 2218. https://doi.org/10.3389/fpsyg.2017.02218

Hitzler, P., 2021. A Review of the Semantic Web Field [WWW Document]. URL https://cacm.acm.org/magazines/2021/2/250085-a-review-of-the-semantic-web-field/fulltext.

Horrocks, I., Patel-Schneider, P.F., 2011. KR and Reasoning on the Semantic Web: OWL, in: Domingue, J., Fensel, D., Hendler, J.A. (Eds.), Handbook of Semantic Web Technologies. Springer, Berlin, Heidelberg, pp. 365–398. https://doi.org/10.1007/978-3-540-92913-0_9

Hsu, M.-H., 2008. A personalized English learning recommender system for ESL students. Expert Systems with Applications 34, 683–688. https://doi.org/10.1016/j.eswa.2006.10.004

Huang, C., Liu, L., Tang, Y., Lu, L., 2011. Semantic Web Enabled Personalized Recommendation for Learning Paths and Experiences, in: Zhu, M. (Ed.), Information and Management Engineering, Communications in Computer and Information Science. Springer, Berlin, Heidelberg, pp. 258–267. https://doi.org/10.1007/978-3-642-24022-5_43

Huang, Z., 1997. CLUSTERING LARGE DATA SETS WITH MIXED NUMERIC AND CATEGORICAL VALUES 14.

Ibrahim, M.E., n.d. An Ontology-based Hybrid Approach to Course Recommendation in Higher Education 167.

Ivezic, N., Schlenoff, C., 2000. ONTOLOGY ENGINEERING FOR DISTRIBUTED COLLABORATION IN MANUFACTURING 7.

J. Sandvig, R. Burke, 2005. AACORN: A CBR Recommender for Academic Advising | BibSonomy [WWW Document]. Tech. Rep. TR05-015. URL https://www.bibsonomy.org/bibtex/ab6c3cb8fdf42fea21eef7d1ab8fd748 (accessed 6.29.20).

Jay, N., Nuemi, G., Gadreau, M., Quantin, C., 2013. A data mining approach for grouping and analyzing trajectories of care using claim data: the example of breast cancer. BMC Medical Informatics and Decision Making 13, 130. https://doi.org/10.1186/1472-6947-13-130

Jhon K. Tarus, Zhendong Niu, Abdallah Yousif, 2017. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. https://doi.org/10.1016/j.future.2017.02.049

Jurafsky, D., Martin, J.H., 2009. Speech and Language Processing (2nd Edition). Prentice-Hall, Inc., USA.

Kang, Y.-B., Krishnaswamy, S., Zaslavsky, A., 2011. Retrieval in CBR Using a Combination of Similarity and Association Knowledge, in: Tang, J., King, I., Chen, L., Wang, J. (Eds.), Advanced Data Mining and Applications, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 1–14. https://doi.org/10.1007/978-3-642-25853-4_1

Kantor, P. B, Rokach, L., Ricci, F., Shapira, B., 2011. Recommender systems handbook (Vol. I). Springer, New York, Dordrecht, Heidelberg, London. https://doi.org/10.1007/978-0-387-85820-3

Khoury, L.A.A., 2020. Transition from High School to University in Lebanon 8.

Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z., 2011. E-Learning personalization based on hybrid recommendation strategy and learning style identification. Computers & Education 56, 885–899. https://doi.org/10.1016/j.compedu.2010.11.001

Kolbert, A.P., 2017. A SCALABLE RECOMMENDER SYSTEM 79.

KOS_Semantic_Digital_Libraries.pdf, n.d.

Kowalski, M., Klüpfel, H., Zelewski, S., Bergenrodt, D., Saur, A., 2013. Integration of Case-Based and Ontology-Based Reasoning for the Intelligent Reuse of Project-Related Knowledge, in: Clausen, U., ten Hompel, M., Klumpp, M. (Eds.), Efficiency and Logistics, Lecture Notes in Logistics. Springer, Berlin, Heidelberg, pp. 289–299. https://doi.org/10.1007/978-3-642-32838-1_31

Levenshtein, V.I., 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady 10, 707.

Liu, A., Callvik, J., 2017. Using Demographic Information to Reduce the New User Problem in Recommender Systems 35.

Lokhande, A., Jain, P., 2019. Hybrid Collaborative Filtering Model Using Hierarchical Clustering and PCA (SSRN Scholarly Paper No. ID 3365525). Social Science Research Network, Rochester, NY. https://doi.org/10.2139/ssrn.3365525

McBride, B., 2002. Jena: a semantic Web toolkit. IEEE Internet Computing 6, 55–59. https://doi.org/10.1109/MIC.2002.1067737

Mota, D., Carvalho, C.V. de, Reis, L.P., 2014. OTILIA — An architecture for the recommendation of teaching-learning techniques supported by an ontological approach. 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. https://doi.org/10.1109/FIE.2014.7044479

Mougouie, B., Richter, M.M., Bergmann, R., 2003. Diversity-Conscious Retrieval from Generalized Cases: A Branch and Bound Algorithm, in: Ashley, K.D., Bridge, D.G. (Eds.), Case-Based Reasoning Research and Development, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 319–331. https://doi.org/10.1007/3-540-45006-8_26

Murtagh, F., 2011. Hierarchical Clustering, in: Lovric, M. (Ed.), International Encyclopedia of Statistical Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 633–635. https://doi.org/10.1007/978-3-642-04898-2_288

Noy, N.F., McGuinness, D.L., 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems Laboratory 32.

Olson, D.L., Delen, D., 2008. Memory-Based Reasoning Methods, in: Olson, D.L., Delen, D. (Eds.), Advanced Data Mining Techniques. Springer, Berlin, Heidelberg, pp. 39–52. https://doi.org/10.1007/978-3-540-76917-0_3

OWL Web Ontology Language Overview, 2020.

OWL Web Ontology Language Overview [WWW Document], n.d. URL https://www.w3.org/TR/owl-features/ (accessed 6.28.20b).

Perner, P., 2019. Case-Based Reasoning – Methods, Techniques, and Applications. pp. 16–30. https://doi.org/10.1007/978-3-030-33904-3_2

Poelmans, J., Elzinga, P., Dedene, G., 2013. Retrieval of criminal trajectories with an FCA-based approach, in: ECIR 2013.

Pukkhem, N., 2014. LORecommendNet: An Ontology-Based Representation of Learning Object Recommendation, in: Boonkrong, S., Unger, H., Meesad, P. (Eds.), Recent Advances in Information and Communication Technology, Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 293–303. https://doi.org/10.1007/978-3-319-06538-0_29

Puntheeranurak, S., Tsuji, H., 2007. A Multi-clustering Hybrid Recommender System, in: 7th IEEE International Conference on Computer and Information Technology (CIT 2007). Presented at the 7th IEEE International Conference on Computer and Information Technology (CIT 2007), pp. 223–228. https://doi.org/10.1109/CIT.2007.54

Qiyan Han, Feng Gao, Hu Wang, 2010. Ontology-based learning object recommendation for cognitive considerations, in: 2010 8th World Congress on Intelligent Control and Automation. Presented at the 2010 8th World Congress on Intelligent Control and Automation, pp. 2746–2750. https://doi.org/10.1109/WCICA.2010.5554857

Rani, M., Muyeba, M.K., Vyas, O.P., 2014. A Hybrid Approach Using Ontology Similarity and Fuzzy Logic for Semantic Question Answering, in: Kumar Kundu, M., Mohapatra, D.P., Konar, A., Chakraborty, A. (Eds.), Advanced Computing, Networking and Informatics- Volume 1, Smart Innovation, Systems and Technologies. Springer International Publishing, Cham, pp. 601–609. https://doi.org/10.1007/978-3-319-07353-8_69

Recio-García, J. A., González-Calero, P. A., Díaz-Agudo, B., 2014. Jcolibri2: A framework for building Case-based reasoning systems. Science of Computer Programming 79, 126–145. https://doi.org/10.1016/j.scico.2012.04.002.

Ricci, F., Wöber, K., Zins, A., 2005. Recommendations by Collaborative Browsing, in: Frew, A.J. (Ed.), Information and Communication Technologies in Tourism 2005. Springer-Verlag, Vienna, pp. 172–182. https://doi.org/10.1007/3-211-27283-6_16

Ristoski, P., Paulheim, H., 2016. Semantic Web in data mining and knowledge discovery: A comprehensive survey. Web Semantics: Science, Services and Agents on the World Wide Web 36, 1–22. https://doi.org/10.1016/j.websem.2016.01.001

Rodriguez, M.Z., Comin, C.H., Casanova, D., Bruno, O.M., Amancio, D.R., Costa, L. da F., Rodrigues, F.A., 2019. Clustering algorithms: A comparative approach. PLOS ONE 14, e0210236. https://doi.org/10.1371/journal.pone.0210236

Rodríguez, P.A., Ovalle, D.A., Duque, N.D., 2015. A Student-Centered Hybrid Recommender System to Provide Relevant Learning Objects from Repositories, in: Zaphiris, P., Ioannou, A. (Eds.), Learning and Collaboration Technologies, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 291–300. https://doi.org/10.1007/978-3-319-20609-7_28

Ruchika, Singh, A.V., Sharma, D., 2015. Evaluation criteria for measuring the performance of recommender systems, in: INFOCOM 2015. https://doi.org/10.1109/ICRITO.2015.7359280

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S., 2007. Collaborative Filtering Recommender Systems, in: Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 291–324. https://doi.org/10.1007/978-3-540-72079-9_9

Semantic Web - W3C [WWW Document], n.d. URL https://www.w3.org/standards/semanticweb/ (accessed 6.28.20).

Serhiy A. Yevtushenko, 2000. System of data analysis “Concept Explorer.” Proceedings of the 7th national conference on Artificial Intelligence KII-2000, p. 127-134, Russia.

Shen, L., Shen, R., 2005. Ontology-based learning content recommendation. https://doi.org/10.1504/IJCEELL.2005.007719

Shishehchi, S., Banihashem, S.Y., Zin, N.A.M., Noah, S.A.M., 2012. Ontological Approach in Knowledge Based Recommender System to Develop the Quality of E-learning System 9.

Silva, P.R.C., Dias, S.M., Brandão, W.C., Song, M.A., Zárate, L.E., 2017. Formal Concept Analysis Applied to Professional Social Networks Analysis:, in: Proceedings of the 19th International Conference on Enterprise Information Systems. Presented at the 19th International Conference on Enterprise Information Systems, SCITEPRESS - Science and Technology Publications, Porto, Portugal, pp. 123–134. https://doi.org/10.5220/0006333401230134

Siri, A., Bragazzi, N.L., Khabbache, H., Spandonari, M.M., Cáceres, L.A., 2016. Mind the gap between high school and university! A field qualitative survey at the National University of Caaguazú (Paraguay). Adv Med Educ Pract 7, 301–308. https://doi.org/10.2147/AMEP.S103811

Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y., 2007. Pellet: A Practical OWL-DL Reasoner (SSRN Scholarly Paper No. ID 3199351). Social Science Research Network, Rochester, NY. https://doi.org/10.2139/ssrn.3199351

Škopljanac-Mačina, F., Blašković, B., 2014. Formal Concept Analysis – Overview and Applications. Procedia Engineering 69, 1258–1267. https://doi.org/10.1016/j.proeng.2014.03.117

Staab, S., Studer, R. (Eds.), 2009. Handbook on Ontologies, 2nd ed, International Handbooks on Information Systems. Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92673-3

Staab, S., Studer, R. (Eds.), 2004. Handbook on Ontologies, International Handbooks on Information Systems. Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24750-0

Szczepaniak, P.S., Duraj, A., 2018. Case-Based Reasoning: The Search for Similar Solutions and Identification of Outliers. Complexity 2018, e9280787. https://doi.org/10.1155/2018/9280787

Tartir, S., Arpinar, I., Moore, M., Sheth, A., Aleman-Meza, B., 2005. OntoQA: Metric-Based Ontology Quality Analysis. Kno.e.sis Publications.

Tarus, J., Niu, Z., Khadidja, B., 2017. E-Learning Recommender System Based on Collaborative Filtering and Ontology. International Journal of Computer and Information Engineering 11, 256–261.

Tarus, J.K., Niu, Z., Yousif, A., 2017. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. Future Generation Computer Systems 72, 37–48. https://doi.org/10.1016/j.future.2017.02.049

Teaching as a research-based profession: Possibilities and prospects (The teacher training agency lecture 1996) | BibSonomy [WWW Document], n.d. URL https://www.bibsonomy.org/bibtex/1a10667879510747ea78631f057e2336a/yish (accessed 8.30.21).

Tett, L., Cree, V.E., Christie, H., 2017. From further to higher education: transition as an on-going process. High Educ 73, 389–406. https://doi.org/10.1007/s10734-016-0101-1

Towle, B., Quinn, C.N., 2000. Knowledge Based Recommender Systems Using Explicit User Models [WWW Document]. undefined. URL /paper/Knowledge-Based-Recommender-Systems-Using-Explicit-Towle-Quinn/efc65009adfb266be054d3e9fea26f57a4019874 (accessed 6.29.20).

Tsarkov, D., Horrocks, I., 2006. FaCT++ Description Logic Reasoner: System Description, in: Furbach, U., Shankar, N. (Eds.), Automated Reasoning, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 292–297. https://doi.org/10.1007/11814771_26

Uschold, M., Gruninger, M., 1996. Ontologies: principles, methods and applications. The Knowledge Engineering Review 11, 93–136. https://doi.org/10.1017/S0269888900007797

Vall, A., Dorfer, M., Eghbal-zadeh, H., Schedl, M., Burjorjee, K., Widmer, G., 2019. Feature-combination hybrid recommender systems for automated music playlist continuation. User Modeling and User-Adapted Interaction 29, 527–572. https://doi.org/10.1007/s11257-018-9215-8

Varga, V., Jánosi-Rancz, K.T., Kalman, B., 2016. Conceptual Design of Document NoSQL Database with Formal Concept Analysis. Acta Polytechnica Hungarica 13, 229–248.

Victoria Shannon, 2006. A “more revolutionary” Web. International Herald Tribune.

Vlaardingerbroek, B., Al-Hroub, A., Saab, C., 2017. The Lebanese Education System. pp. 255–265. https://doi.org/10.1007/978-94-6300-992-8_16

V.N., G., 2007. The Undecided College Student: An Academic and Career Advising Challenge. Charles C Thomas Pub Ltd.

Vozalis, E., Margaritis, K.G., 2007. Analysis of Recommender Systems’ Algorithms 15.

Vozalis, M.G., Margaritis, K.G., 2007. Using SVD and demographic data for the enhancement of generalized Collaborative Filtering. Inf. Sci. 177, 3017–3037. https://doi.org/10.1016/j.ins.2007.02.036

W3C, 2004a. RDF/XML Syntax Specification (Revised).

W3C, 2004b. OWL Web Ontology Language Overview, W3C Recommendation.

Wong, W., Liu, W., Bennamoun, M., 2012. Ontology learning from text: A look back and into the future. ACM Comput. Surv. 44, 20:1-20:36. https://doi.org/10.1145/2333112.2333115

Xia, W., He, L., Gu, J., He, K., 2009. Effective Collaborative Filtering Approaches Based on Missing Data Imputation, in: 2009 Fifth International Joint Conference on INC, IMS and IDC. Presented at the 2009 Fifth International Joint Conference on INC, IMS and IDC, pp. 534–537. https://doi.org/10.1109/NCM.2009.128

Zarzour, H., Al-Sharif, Z., Al-Ayyoub, M., Jararweh, Y., 2018. A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques, in: 2018 9th International Conference on Information and Communication Systems (ICICS). Presented at the 2018 9th International Conference on Information and Communication Systems (ICICS), IEEE, Irbid, pp. 102–106. https://doi.org/10.1109/IACS.2018.8355449

Zhang, L., 2013. The Definition of Novelty in Recommendation System. Journal of Engineering Science and Technology. https://doi.org/10.25103/jestr.063.25

Zhuhadar, L., Nasraoui, O., 2010. A Hybrid Recommender System Guided by Semantic User Profiles for Search in the E-learning Domain. JETWI 2, 272–281. https://doi.org/10.4304/jetwi.2.4.272-281

APPENDIXES

Appendix A

Semantic web, Ontology and case-based reasoning

This appendix introduces the *Semantic Web (SW)* technologies and ontology, and an overview on the CBR systems. In the following section, an introduction about the *SW* is presented.

Introduction

Petabytes of data are published online and available for all internet users. However, these data are not understandable by the machine. Besides, the HTML pages present the online data in many different formats that are difficult for machines to process. Here comes the major role of the *SW*, which provides a solution for the machine to treat online data. The *SW* (Frauenfelder, M., 2004) is an extension of the current *World Wide Web* (WWW) that offers programmable application with machine-interpretable metadata of the online data. The *SW* adds extra data descriptors to available content on the Web. This enables machines to make meaningful interpretations the same way people analyze data to make useful decisions. The motivations for applying the *SW* technologies to the Web comes under the umbrella of three factors: Individual Assistants, Automated Information Retrieval (AIR), and the Internet of Things (IoT). Thus, the *SW* represents the next major evolution in linking knowledge through semantic relationships. These relationships enable Web data to be linked and understood by machines in order to compute sophisticated tasks on users' behalf. So, the Web is directed toward a new phase of development.

The essential difference between the *SW* and other technologies that deal with data such as the *WWW* or databases is that the *SW* is interested in the meaning of data more than in its structure. The *SW* consists of three main technical standards: *Resource Description Framework (RDF)* (W3C, 2004a), *Web Ontology Language (OWL)* (W3C, 2004b), *SPARQL Protocol and RDF Query Language (SPARQL).* The following figure shows the evolution phases of the Web.

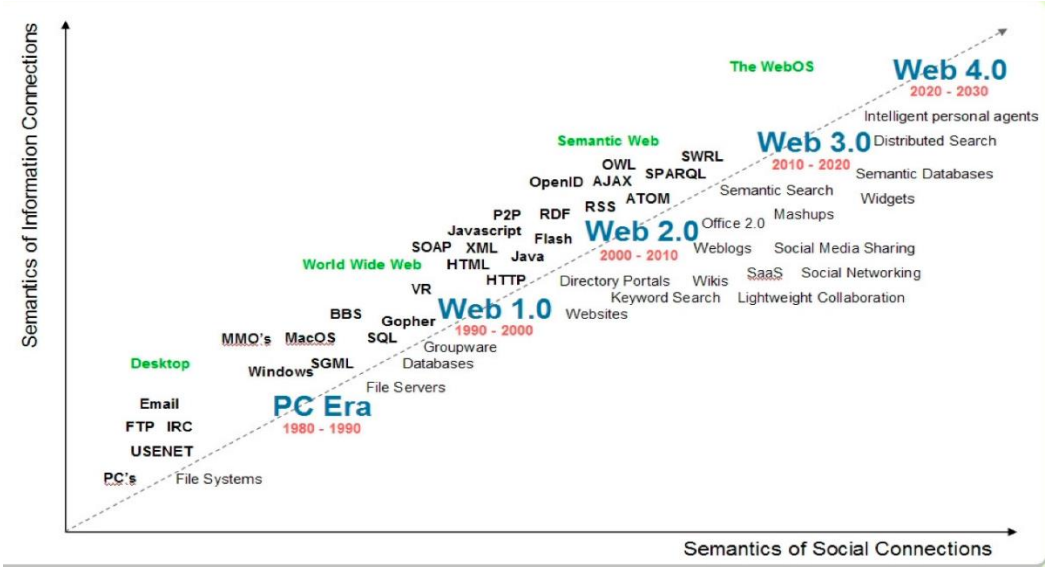


Figure 1: Evolution of the Web (source: Radar Networks & Nova Spivack 2007)

With the *SW* technologies, *Web 3.0* will be more intelligent, linked, and open. The following section presents a brief description of the Web 3.0.

The web 3.0

The *Web 3.0* (Victoria Shannon, 2006) is the reinvention of the *Web 2.0*. The latter allows users to interact with dynamic webpages, which act as applications and not as simple static webpages. In the *Web 2.0* users can use search engines such as *Google* to search for information, which provides satisfactory returns. This search engine processes the search for keywords and does not understand the semantic of the search. For example, if a user searches for an animal named *Jaguar*, then 90 percent of the search returns are for the *Jaguar car* because this car is the most popular search result.

On the other hand, the *Web 3.0* can get the context of the keyword from the internet user in order to find the most useful information about the *Jaguar animal*. The *Web 3.0* can be linked to an artificial intelligence application that understands users' behavior and interests. Additionally, *Web 3.0* search engines will be able to present the information to internet users in an intelligent way, even provide accurate and personalized search results. Furthermore, The *Web 3.0* can make computers intelligent by making them capable of processing and understanding online data. Thus, the *Web 3.0* is going to transform the websites into Web services.

Finally, The *Web 3.0* will be linked and working with the *SW*, *Natural Language Processing (NLP), Machine Learning (ML) and Reasoning*, *Distributed Databases,* etc. In the next section, the *SW* applications, *SW* standards, and *SW* challenges and limitations are presented.

The semantic web

The *SW* has been planned by Tim Berners-Lee as an extension of the recent Web in order to facilitate human-computer cooperation (Domingue, J. et al., 2011). Machines are not currently capable of understanding or interpreting Web content. However, the *SW* permits machines to understand Web content by themselves (Frauenfelder, M., 2004). It can be considered as a global data network. This *SW* allows data to be shared and reused across communities, enterprises, and application boundaries. The *SW* purpose is to transform the existing Web, limited by unstructured and semi-structured data into *Web of data*. The Web should define links between data items such as *illustrates, encompasses, composes*, etc. Unluckily, the relationship between the data items is not popular on the Web. However, the technology to achieve such relationships is available, and is named RDF.

The World Wide Web Consortium (W3C) (W3C, 2004b) helped in building the *SW* that support the *Web of data*. The aim of the *Web of data* is to allow machines to do useful and intelligent work. The *SW* enables internet users to create Web stores for storing *linked data* and building

vocabularies. Technologies such as *RDF*, *OWL*, *Extensible Markup Language* (*XML*) (Fennell, Philip., 2013), and *SPARQL* are used in order to empower Linked Data.

Semantic web Applications

The *SW* technologies are widely used in many Web applications such as:

- *Semantics-based Search Engines,* which are defined by ontology such as *Swoogle*[8].
- *Agent-based Distributed* (Gorshkov, 2015) *Applications,* which allow the use of data that is structured, defined, and interpreted by ontologies.
- *Ontology-based Information Management Systems* (Hailemariam et al., 2006)*,* which allow enterprises to manage information effectively.
- *Semantics-Based Digital Libraries*("KOS_Semantic_Digital_Libraries.pdf," 2012)*,* which provide effective indexing and classification of information in order to access digital libraries easier.

Semantic web challenges and limitations

Ontologies are the carriers of the meaning involved in the *SW*. Therefore, ontologies are key to the *SW* that offer semantics and vocabulary of the annotations. However, ontologies raise major challenges when it comes to their construction and adaptation:

- The *construction* issue is related to building the core ontologies to be reused by all domains. In addition, the process for the construction of ontology for a specific domain is complex and hard to be applied. The difficulty of constructing domain ontologies is related to the complexity of the domain knowledge such as medicine, education, tourism, and genetics.
- How to extend and update the existing ontologies is the *adaptation* issue*,* since ontologies evolve over time. This issue encompasses reasoning and editing ontologies and searching for them in a library.

Designing the *SW* content is a serious challenge for the *SW*. In addition, the *SW* has many other limitations (Ristoski and Paulheim, 2016) such as *vastness, vagueness, uncertainty, inconsistency,* and *deceit*. Any *Automated Reasoning* application should deal with these limitations in order to attain the *SW* purposes.

Vastness: The *SNOMED CT* medical ontology alone has 370,000 class names. Additionally, many duplicated semantic terms exist online . Therefore, *Automated Reasoning* applications should deal with huge inputs.

Vagueness: This issue can arises when people want to agree on the precise meaning of some terms and the machines want to reason with them.

Uncertainty: This issue arises when there are precise concepts with uncertain (inexact) values. Also, uncertainties are caused, for example, by lack of knowledge about the domain, and unreliable sources of information.

[8] http://swoogle.umbc.edu/2006/

Inconsistency: This issue arises when ontologies are acquired from separate sources and formally contradict each other.

Deceit: this issue occurs when the publisher of the information misleads the internet user intentionally.

In the next section, we focus on the notion of ontology and the Web Ontology Language OWL, its features and syntax, ontology structure, ontology engineering, ontology reasoning, ontology evaluation.

Ontologies

In 1993, Gruber (Gruber, 1993) has defined the concept of ontology as an "explicit specification of a conceptualization". An ontology is a formal description of knowledge as a collection of classes within a specific domain and the relations that hold between them (Staab and Studer, 2004). Recently, ontologies have attracted extensive attention in designing domain knowledge of courses, e-learning, news, software engineering, etc. The main ontology components are classes (sets of objects), instances (objects/individuals), properties (binary relations between objects), and axioms (semantic constraints on the former). These components are used to design the object-oriented model for specific domain knowledge and share it for reuse on the Web. Moreover, an ontology enables Web content to be understandable by humans and machines alike.

The benefit of using ontologies is that, by describing the relations between concepts constructed into them, they allow automated reasoning about knowledge. Ontologies also offer the value that they are agnostic to data formats such as structured, unstructured, or semi-structured data. This makes them usable to support data integration, data analysis, and concept and text mining.

OWL features and syntax

Ontology consists of terminological knowledge (*Tbox*) and assertional knowledge (*Abox*).

The *Tbox* defines class and properties. The following are the main parts of *Tbox:*

- *Class hierarchy*: classes are related to each other by subsumtion (inclusion) relations.
- *Object properties*: Binary relations between instances, e.g. *loves* (as in *rami loves cathy*).
- *Data properties*: Relations between instances to values, e.g. *hasWeight* (as in *rami hasWeight "75.4"^^xsd:float*).

The Abox: Defines concrete instances and their connections to other instances and values.

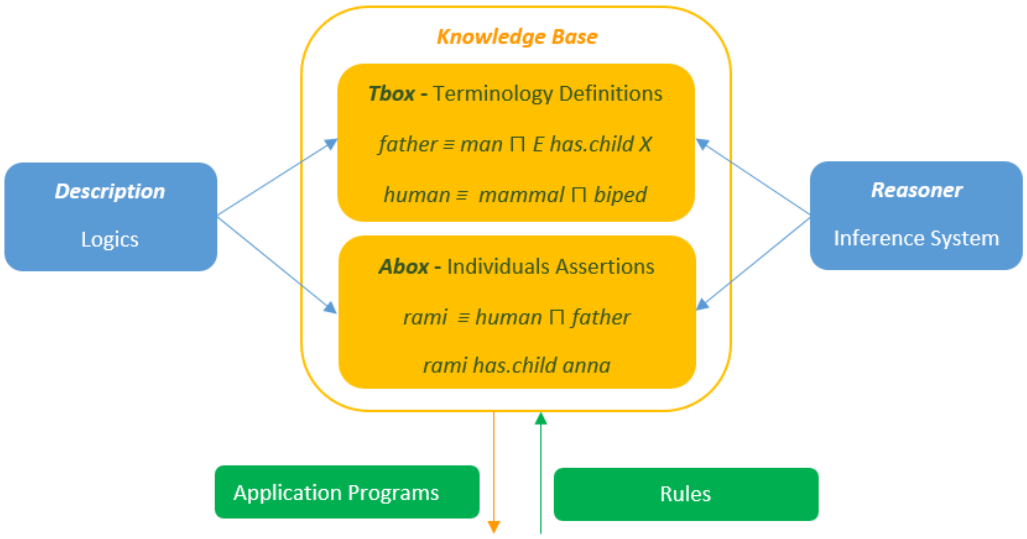


Figure 4: Description Logics Architecture

OWL is based on *Descriptive Logic (DL)* that provides a set of *constructors* and *axioms* for building ontologies. *DL* is a family of logic-based knowledge representation formalisms, which describes domain knowledge in terms of concepts/classes, roles/relationships, and individuals.instances. *OWL* uses axioms for identifying disjoint, equivalence, subsumption, and property characteristics.

The traditional *DL* syntax is used for logical expressions and is not well understood by non-logicians. Therefore, the *Manchester DL* syntax was declared as a user-friendly syntax in order to be used in software such as *Protégé* ontology editor.

OWL Constructor	***Traditional DL Syntax***	***Manchester OWL Syntax***	***Example***
IntersectionOf	C ⊓ D	C AND D	Human AND Male
UnionOf	C ⊔ D	C OR D	Man OR Women
ComplementOf	¬ C	NOT C	NOT Male
SomeValuesFrom	∃R.C	R SOME C	hasColleague SOME Teacher
AllValuesFrom	∀R.C	R ONLY C	hasColleague ONLY Teacher
MinCardinality	≥ n R	R MIN n	hasColleague MIN 4
MaxCardinality	≤ n R	R MAX n	hasColleague MAX 4
Cardinality	= n R	R EXACTLY n	hasColleague EXACTLY 4
HasValue	∃R.{x}	R VALUE x	hasColleague VALUE rami

Table 1: Traditional DL vs Manchester Syntax

In *Manchester DL*, mathematical symbols such as (∃,∀, and ¬) have been substituted by natural keywords such as (some, only, and not).

As an example, the below axioms define the knowledge about burgers:

Axioms	
Axioms (a) and (b) describes that MexicanBurger is a Burger, and has MexicanFlavor.	MexicanBurger ⊑ Burger (a) MexicanBurger ⊑ ∃ hasSauce.MexicanFlavor (b)
Axiom (c) describes that MexicanFlavor is a SauceFlavor.	MexicanFlavor ⊑ SauceFlavor (c)
Axiom (d) describes that SauceyBurger is precisely those Burger that has SauceFlavor.	SauceyBurger ≡ Burger ⊓ ∃ hasSauce.SauceFlavor (d)

Table 2: Axioms Example

We deduce from table above that the concept Mexican Flavor is a subclass of SauceyBurger, as it is a Burger, and has MexicanFlavor.

Ontology engineering

Ontology engineering is a process implemented for the development of an ontology for a specific domain (Ivezic and Schlenoff, 2000). Ontology engineering is a field of knowledge engineering that focuses on the ontology development process, ontology life cycle, ontology design methodologies, and ontology tools and languages (Gómez-Pérez et al., 2004).

Grimm et al. (Grimm et al., 2011) specified a generic method of ontology engineering that involves three essential phases:

Requirements analysis: the domain experts study the requirements of the application scenario, and then define them as ontology requirement in the specification and description file. This file should comprise information about the scope and level of expressivity of the ontology.

Conceptualization: the domain ontological entities and the axioms are presented in terms of a semantic vocabulary. Domain experts build the structure of the ontology that fit the requirements descriptions and specifications. According to Uschold and Gruninger (Uschold and Gruninger, 1996), there are three approaches to design the conceptual model relationships or class hierarchy:

- A *top-down* approach, where the design procedure begins from the top general concepts and then classifies the subsequent knowledge of these concepts.
- A *bottom-up* approach, where the design procedure begins from the most specific concepts as the leave nodes in the concept hierarchical structure, and then groups it into concepts that are more general.
- A *hybrid* approach combines the top-down and bottom-up techniques.

Implementation: in this phase, the ontology language formalizes the specifications and some automated ontology acquisition or reuse methods might be used.

Besides, another ontology engineering process is used namely the "*Ontology Development 101*" (Noy and McGuinness, 2001) that provides an organized overview of necessary steps in an

ontology engineering project. This process does not need any past knowledge of ontology theory or experience. Also, it provide

real and applicable direction on practical issues of ontology engineering. The following figure illustrates the *Ontology Development 101* process.

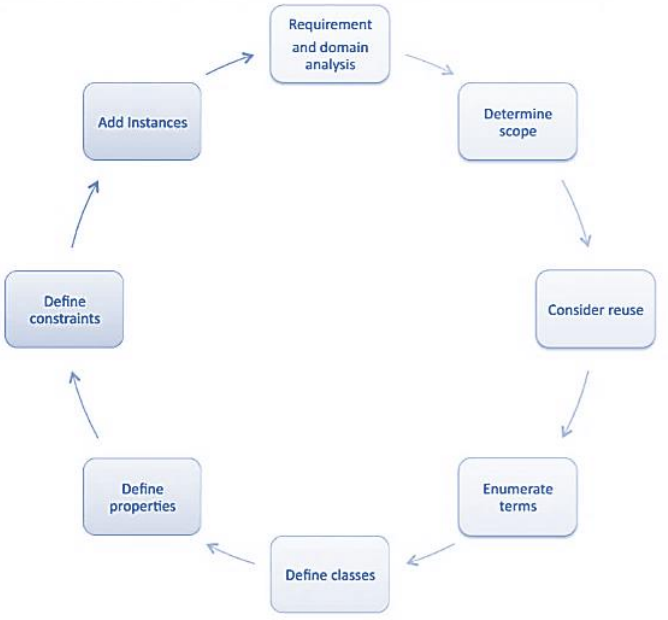


Figure 5: Ontology Development Process

Phase 1: *Determine the domain and scope of the ontology*

In this phase, the development of the ontology process starts by defining:

- The domain that the ontology will describe.
- The usage purpose of the ontology.
- The sorts of questions the ontology will answers.
- The users who will use and maintain the ontology.

Phase 2: Consider reusing existing ontologies
It is important to consider reusing existing ontologies in order to interact with other applications that have already integrated ontologies. Ontologies are available online and can be reused in our ontology development environment.

Phase 3*: Enumerate important terms in the ontology*
In this phase, all the domain terms are described and explained to users. For example, important education-related terms will include *University, School, location, Course, University Major, Academic Degree, etc*.

Phase 4: *Define the classes and the class hierarchy*.
In this phase, the three development approaches presented above (*top-down, bottom-up, or hybrid*) can be used to design the class hierarchy or conceptual model relationships.

Phase 5: *Define the properties of classes*

In this phase the internal structure of concepts is described. Each property should determine which class it describes. Thus, the *Course* will have the following slots: *level, language, prerequisite* and the class *University* will have a *location* slot.

Phase 6: *Define the facets of the slots*
Slots can have different facets:

Slot cardinality: describes the number of values a slot can have. Systems differentiate between the following cardinality:

- Single cardinality that allows maximum one value.
- Multiple cardinality that allows unlimited number of values.
- Minimum cardinality that identifies the minimum number of values a slot can have.
- Maximum cardinality that identifies the maximum number of values a slot can have.

Slot-value type: defines what types of values the slot can have. The following are the common value types:

- *Variable: String, Number and Boolean*.
- *Enumerated:* we can specify that the *level* slot can take on one of the three possible values (beginner, intermediate, and advanced).
- *Instance-type:* slots permit description of relations between instances.

Domain and range of a slot: defines *classes* for slots, which is the range of a slot of type *Instances*. The domain of a slot are the classes to which a slot is linked or the classes' property a slot defines.

Phase 7: *Create instances*
In this phase, instances of classes are created. Three steps to define an instance (choose the class, create the instance of that class and then input the slot values. We can create an instance *computer science* to represent a specific type of *University major*.

Ontology reasoning

A semantic reasoner is a software module used to infer logical expressions from a set of declared axioms. Inference on the *SW* is discovering relationships based on knowledge and vocabularies. Besides, ontology reasoning is a core task used for discovering concepts of ontology and answering domain queries (Horrocks and Patel-Schneider, 2011). Additionally, ontology reasoning can detect conflicts and help eliminate redundancy in the knowledge base. Thus, ontology reasoning generates inferences from available facts by referring to a set of rules that are defined in the axioms of an ontology.

The core reasoning tasks over ontologies are:

Subsumption reasoning: this task infers when a class is a subclass of another. For example, X is a subclass of class Y, so all the individuals/instances of X should be individuals of Y. This is true

due to the explicit assertion or the inference process. Thus, concept hierarchies that represent the conceptual model can be built in this task.

Satisfiability reasoning: this task states if a concept is unsatisfiable, e.g. a concept contains contradictory axioms.

Instance reasoning: this task retrieves the instances of a particular class.

Instantiation: this task retrieves the classes that *n* is an instance of.

The following are some important examples of OWL reasoning tools:

- Pellet (Sirin et al., 2007) is a full OWL-DL reasoner.
- Racer (Haarslev, V. and M¨oller, R., 2001) and FaCT++ (Tsarkov and Horrocks, 2006).

The following figure illustrates some of the available *SW* tools.

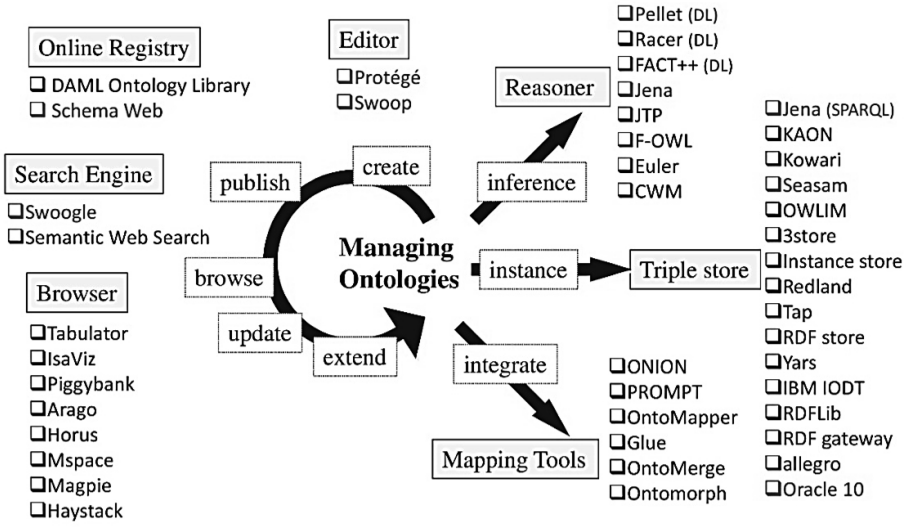


Figure 6: Semantic Web Tools[9]

Figure 6 shows the *SW* tools used to support the SW technologies, which encompasses the *Online Registry, Ontology Editor, Ontology Reasoner, Ontology Search Engines, Ontology Browsers, Triple Stores, and Mapping Tools.*

Ontology evaluation

It is a hard and complex task to evaluate if an ontology is good, not good.
There are two types of ontology evaluation namely ontology *verification* and *validation* (Bilgin et al., 2014).

- *Ontology verification*: this type verifies if the ontology is built correctly and ensures if its descriptions apply properly to the requirements.
- *Ontology validation*: this type validates if the meaning of the definitions represents the domain for which the ontology was made.

[9] https://www.w3.org/standards/semanticweb/

The evaluation must be done for every ontology aspect such as vocabulary, structure, semantics, syntax, representation, and context (Staab and Studer, 2009). However, there are many available techniques for ontology evaluation. Thus, it is very important to select an appropriate ontology evaluation technique that fits the domain. *Tartir et ao*. (Tartir et al., 2005) proposed a set of metrics to analyze ontology schemata and their populations. Among them, "*Schema metrics"* and "*Instance metrics"* are used for the evaluation of ontology construction.

Finally, the *SW* will be the Web of the next generation. However, many issues, limitations, and challenges are facing this technology such as semantic interconnection and reasoning, theoretical barriers, and technical obstacles. In the next sections, an overview on the CBR systems is presented.

Case-based reasoning systems

Case-based reasoning (CBR) (Perner, 2019) is an *Artificial Intelligence* (AI) technique applicable to problem-solving and learning where earlier cases are available. CBR is the process of addressing a new problem based on the solutions of similar prior problems. Solutions for the current problem are retrieved from a library of prior cases called case-base. Researchers revealed that most people collect solutions based on previous experiences with similar cases. For example, a vehicle mechanic who repairs a car engine using his experience with another car that showed similar symptoms is implementing CBR.

The automated reasoning (Das et al., 2021) technique of the CBR defines the problem for the current case, searches the case-base in order to retrieve the most similar prior cases, uses the solutions of the retrieved cases and adapts it to the current case problem, and finally updates the case-base by saving the new experience. Thus, CBR performs as memory and recall is done based on the similarity retrieval and reuse of the most similar solutions. Current addressed problems may be retained and the memory grows as problem-solving happens. Besides, CBR enables the utilization of specific knowledge of prior experienced real cases rather than using only the general knowledge of a problem domain.

Attaining problem-solving, planning, memory, and learning, CBR offers a basis for innovative technology of advanced computer systems that can resolve situations and adapt to new cases. The CBR has the ability to deal with complex real world cases.

Essentials of case-based reasoning methods

CBR encompasses a range of techniques for retrieving, indexing, and organizing the retained knowledge in the case-base. Cases may illustrate the generalized cases that are the set of similar cases and may be retained as real experiences. CBR cases may be saved in the case-base as distinct knowledge entities or segmented into sub-entities and disseminated within the knowledge

construction. Also, CBR cases may be indexed by an open or predefined vocabulary inside a hierarchical or flat structure. The solution of the prior case may be reused as is to solve the current problem, or adjusted according to the dissimilarities between the active case and previous case. CBR techniques may cooperate with the user for guidance of its selections. Some CBR techniques are based on a limited set of cases or on great volume of extensively distributed ones. In CBR systems, the prior cases may be retrieved and assessed in parallel or sequentially. The clarification of the terms related to CBR systems are given below.

Exemplar-based reasoning: A concept can be described extensionally as a set of its exemplars(Goštautaitė and Kurilov, 2021). Exemplar-based CBR methods can address the learning of concept descriptions. In this method, resolving a situation is a *classification function*, for example retrieving the correct class for the uncategorized exemplar. In this method, the class of the most similar prior case is suggested as a solution to the classification problem. A set of classification can create a set of probable solutions. Therefore, the adjustment of a retrieved solution is done outside the scope of the exemplar method.

Instance-based reasoning: This method is a specialization of the exemplar-based reasoning (Dubois et al., 2002). It simply store the existing training data. Once a new instance is faced, the set of similar instances is retrieved from the memory and applied to classify the query instance. Instance-Based Learning is called Lazy Learning since they delay the processing time awaiting a new instance to be classified.

Memory-based reasoning: This method emphasizes a pool of cases as a large memory, and processing by accessing and searching in this memory. The focus of this case-based method is the memory access and organization. What differentiates this method from other approaches is the deployment of the *parallel processing* techniques. Besides, the log on and storing techniques rely on syntactic criteria or utilize general domain knowledge(Olson and Delen, 2008).

Typical case-based reasoning: The characteristics of the case-based reasoning methods are different from the other listed approaches. A classic case is typically expected to have some degree of knowledge richness, and some complexity with respect to its interior structure. The characteristics property of a typical case-based techniques are to *modify* and *adapt* a retrieved solution when implemented in a problem solving environment. Case-based methods uses the overall domain *background knowledge.* However, the degree of explicit representation, richness, and function inside the CBR cycle differs among systems.

Analogy-based reasoning: Typical case-based techniques work on matching and indexing strategies for the cases from a single domain. Analogy-based reasoning focuses on the identification and utilization of cross-domain analogies (Bartha, 2019). In this method, the *mapping problem* is the *reuse* of a prior case: transferring or mapping the solution of a specified analogue, called the *source* to the current problem, called the *target*.

Appendix B

Survey's attributes

Your educational trajectory

* Required

Please select your preferred language to answer our survey. *

Mark only one oval.

- English
- French

Survey description

Selecting a major and a university is a challenging process rife with anxiety. Students at the school are not sure how to match their interests with their working future or major.

In the context of a collaboration between the Lebanese University and Universté Claude Bernard Lyon1 (France), we are conducting a survey to learn about people's interests, problems and satisfaction about their high school, university, university major and future career. The goal is to build a recommender system to provide high-school students with academic advising and guidance.

This survey is totally anonymous and your answers will not be used for any other purpose than the study described above. Please take some time (approximately 5 minutes) to fill it out.

YOUR PERSONAL INFORMATION

Enter your email address. (Optional)

_____

What is your Gender? *

Mark only one oval.

- Male
- Female

What is your age? *

Example: 19, 21, 35...

_____

What is your nationality? *

Multiple answers are allowed

Check all that apply.

- Lebanese
- French

Other: _____

Select your hobby: *

Mark only one oval.

- Animal care
- Bicycling
- Reading
- Camping
- Computer
- Cooking
- Dancing
- Diving
- Drawing
- Family time
- Fashion
- Fishing
- Gardening
- Going to movies
- Hiking
- Music
- Photography
- Sewing
- Shopping
- Skiing
- Sports
- Swimming
- Traveling
- Watching TV
- Writing
- Crafts
- Social
- None
- RidingHorsebk
- Hunting
- Painting

Select the work of your father:

Mark only one oval.

- None
- Agriculture, Food, and Natural Resources. Architecture
- and Construction
- Arts, Audio/Video Technology, and Communications
- Business, Management, and Administration
- Education and Training
- Finance
- Government and Public Administration
- Health Science
- Hospitality and Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections, and Security
- Manufacturing
- Marketing, Sales, and Service
- Science, Technology, Engineering, and Mathematics
- Transportation, Distribution, and Logistics

Select the work of your mother: *

Mark only one oval.

- None
- Agriculture, Food, and Natural Resources Architecture
- and Construction
- Arts, Audio/Video Technology, and Communications
- Business, Management, and Administration Education
- and Training
- Finance
- Government and Public Administration
- Health Science
- Hospitality and Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections, and Security

Manufacturing

Marketing, Sales, and Service

Science, Technology, Engineering, and Mathematics

Transportation, Distribution, and Logistics

Do you take as a model a member of your family? *

Mark only one oval.

Yes

No

YOUR HIGH SCHOOL OR VOCATIONAL SCHOOL INFORMATION

What high school did you attend? *

Multiple answers are allowed

Check all that apply.

Private school

Public school

Do you have a high school degree or equivalent certificate? *

Mark only one oval.

Yes

No

If yes, what is the type of the degree you graduate with from High School?

Multiple answers are allowed

Check all that apply.

Technical

General

Other: _____

What was the name of your high school? *

Please indicate your schools' name (separated by commas if more than one).

_____

How would you rate your high school level on the following?

In this section, please rotate your device to landscape to see all options.

Humanities *

Mark only one oval per row.

	Very Good	Good	Poor
Literature			
Philosophy Religion			

Fine Arts *

Mark only one oval per row.

	Very Good	Good	Poor
Music			
Theatre			
Dance			
Drawing			

Science *

Mark only one oval per row.

	Very Good	Good	Poor
Biology Chemistry			
Physics			
Mathematics			
Science of engineering			

History/Social Sciences *

	Very Good	Good	Poor
History Geography			
Economics			
Sociology			
Psychology			

Languages *

Mark only one oval per row.

	Very Good	Good	Poor
Arabic language English			
language French			
language Other foreign			
language			

Other courses *

Mark only one oval per row.

	Very Good	Good	Poor
Technology and Computer Science			

Physical education (sports)			

When did you graduate from high school? *

Example: 1998, 2010, 2018...

What high school subject did you like best? *

Mark only one oval.

- Literature
- Philosophy
- Religion Music
- Theatre Dance
- Drawing
- Biology
- Chemistry
- Physics
- Mathematics
- History
- Geography
- Economics
- Sociology
- Psychology
- Arabic language English
- language French
- language Other foreign
- language
- Technology and Computer Science
- Physical education (sports)

What high school subject did you like least? *

Mark only one oval.

- Literature
- Philosophy
- Religion Music
- Theatre Dance
- Drawing
- Biology
- Chemistry
- Physics
- Mathematics
- History
- Geography
- Economics
- Sociology
- Psychology
- Arabic language English
- language French
- language Other foreign
- language

How helpful was the school orientation to choose your current university major? *

Mark only one oval.

- Very helpful
- Somewhat helpful
- Not so helpful

YOUR FIRST UNIVERSITY INFORMATION

These questions concerns the university/institution where you succeed your first degree after high school

From which university/institution did you get your first degree after high school? *

Please indicate your university's name

_____

What was your major? *

Example: Computer Science, Accounting...

_____

What was your university's languages of study? *

Multiple answers are allowed

Check all that apply.

- [ ] English
- [ ] French
- [ ] Arabic

How likely are you to recommend this university to others? *

Mark only one oval.

- ◯ Very likely
- ◯ Somewhat likely
- ◯ Not so likely

How effective was the teaching within your major at the university? *

Mark only one oval.

- ◯ Very effective
- ◯ Somewhat effective
- ◯ Not so effective

How would you rate the tuition of your university? *

Mark only one oval.

- ◯ Expensive
- ◯ Fair Cheap

How important were these criteria when you chose your major/university? *

Mark only one oval per row.

	Very important	Moderately important	Not so important
Financial criterion (Registration fees, quality of life, housing, …)	◯	◯	◯
Geographical criterion (big city, distance home-university, …)	◯	◯	◯
Reputation (rank of university,			

university-company relationships, …)			
Family pressure (my parents want that)			
Future employment opportunities			

Did you have a scholarship during your studies at university? *

Mark only one oval.

- Yes
- No

Did you work during your studies at university? *

Mark only one oval.

- Yes
- No

Occupation

Are you currently employed or have a business? *

Mark only one oval.

- Yes *Skip to question 35.*
- No

Education situation

Are you pursuing your studies? *

Mark only one oval.

- Yes *Skip to question 47.*
- No *Skip to question 50.*

YOUR INTEREST AND CAREER INFORMATION

What is the highest level of education you have completed? *

Mark only one oval.

- Bachelor degree
- Master degree
- Doctoral Degree

What kind of job/career interests you? *

Mark only one oval.

- Agriculture, Food, and Natural Resources Architecture
- and Construction
- Arts, Audio/Video Technology, and Communications
- Business, Management, and Administration Education
- and Training
- Finance
- Government and Public Administration
- Health Science
- Hospitality and Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections, and Security
- Manufacturing
- Marketing, Sales, and Service
- Science, Technology, Engineering, and Mathematics
- Transportation, Distribution, and Logistics

Choose the category of your current job *

Mark only one oval.

- Agriculture, Food, and Natural Resources Architecture
- and Construction
- Arts, Audio/Video Technology, and Communications
- Business, Management, and Administration Education
- and Training
- Finance
- Government and Public Administration
- Health Science
- Hospitality and Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections, and Security
- Manufacturing
- Marketing, Sales, and Service
- Science, Technology, Engineering, and Mathematics
- Transportation, Distribution, and Logistics

What is your current job? *

Example: Mechanical Engineer, Accountant...

_____

Is your current job related to your university major? *

Mark only one oval.

- Yes
- No
- Somehow

Did you find difficulties to find your current job? *

Mark only one oval.

- Yes
- No

In how many years did you find your job after your graduation? *

Mark only one oval.

- In less than one year
- Between 1 and 3 years
- More than 3 years

Did you find your job through your network (family, professional, university)? *

Mark only one oval.

- Yes
- No

How satisfied or dissatisfied are you with your current salary? *

Mark only one oval.

- Somewhat satisfied
- Neither satisfied or dissatisfied
- Somewhat dissatisfied

Are you looking to leave your current Job? *

Mark only one oval.

- Yes
- No
- Maybe

If yes, why?

Are you pursuing your studies? *

Mark only one oval.

- Yes
- No *Skip to question 50.*

YOUR CURRENT UNIVERSITY MAJOR INFORMATION

What is your current major? *

Please indicate your current major (separated by commas if more than one).

What degree are you currently pursuing? *

Check all that apply.

- Bachelor degree
- Master degree
- Doctoral degree
- Other:

How likely are you to change your current university major? *

Mark only one oval.

- Very likely
- Somewhat likely Not
- so likely

How many times (if any) did you change your major at university (current or before)? *

Mark only one oval.

- Never
- Once
- Twice
- Three times or more

If you already changed your university major, why did you change it?

Multiple answers are allowed

Check all that apply.

- Badly advised

- [ ] Lack of understanding
- [ ] You were uninterested in courses You
- [ ] had new interests
- [ ] You could not see yourself doing the job in the future You
- [ ] had financial circumstances
- [ ] You had personal circumstances
- [ ] You were doing what your family wants
- [ ] Other: _____

How likely are you to recommend your major to others? *

Mark only one oval.

- ◯ Very likely
- ◯ Somewhat likely
- ◯ Not so likely

How well does your major meets your needs or interests? *

Mark only one oval.

- ◯ Very well
- ◯ Somewhat well
- ◯ Not so well