



# **An Ontology-based Hybrid Approach to Course Recommendation in Higher Education**

By

Mohammed Essmat Ibrahim

The thesis is submitted in partial fulfilment of the requirements for the award of the degree  
of Doctor of Philosophy of the University of Portsmouth

School of Energy and Electronic Engineering

Faculty of Technology

University of Portsmouth

January 2019

## **ABSTRACT**

Both the quantity and complexity of course-related information available to students are rapidly increasing on the Web. This potential information overload challenges standard information retrieval models as users find it increasingly difficult to locate relevant information. The education domain is one of the main domains that has been influenced by this problem. Choosing a higher education course at university can be incredibly tedious and extremely complicated for students. A personalised recommendation system can be an effective way of suggesting relevant courses to prospective students. The existing methods which are mainly based on keywords fail to address the individual user's needs in the recommendation process. Although models use collaborative filtering there is often a lack of historical information. Another shortcoming is that they do not provide comprehensive knowledge about the course that is most relevant to the student.

This research presents a novel ontology-based hybrid approach to recommend personalised courses to match student's individual needs by integrating all available information about courses and supporting students to choose courses towards their career goals. This thesis makes three major contributions: firstly, it proposes a comprehensive Ontology based Personalised Course Recommendation framework, called OPCR, by combining several artificial intelligence techniques including collaborative filtering, content-based filtering, ontological representation and management of knowledge. A set of ontology based recommendation algorithms are developed for personalised recommendation. The framework is capable of automatic data extraction, integration to provide students with suitable recommendations to meet their needs. It not only reduces information overloading but also improves recommendation accuracy. Secondly, it proposes ontology models to extract and integrate information from multiple sources, which contributes to improving the quality of the recommendations by overcoming the heterogeneity of course information. In addition, it has properties such as generality which enables it to be used in different recommendation system domains which change with the user's interests and the object's attributes. Finally, a personalised recommendation system based on the OPCR framework is developed and evaluated. The system is available online as open access for researchers and developers. Results show that the ontology based recommendation algorithms that use hierarchically related concepts produce better outcomes compared to a filtering method that considers only keyword similarity. In addition, the system's performance is improved when the ontology similarity between the items' profiles and the users' profiles is utilised.

## **DECLARATION**

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

## **ACKNOWLEDGEMENT**

First and foremost, I would like to thank the Creator of the Universe, Most Gracious and Most Merciful, without whose Will, it would not have been possible for me to fulfil my wish of completing this PhD.

I would like to express my gratitude to many people who have inspired me both personally and professionally. First and foremost, I would like to thank my first supervisor, Dr Linda Yang who introduced me to the field of Artificial Intelligence and Information Retrieval Systems. I would never have had the courage to go forward with the methodology and theoretical research without her tremendous confidence in me. I can barely think of anything that has been achieved without her help and she has offered me the freedom to autonomously explore new ideas. Over the last four years, I have been blessed with her guidance, encouragement, tolerance, freedom, trust, hospitality and her friendship.

I would also like to acknowledge the School of Engineering in the Faculty of Technology at the University of Portsmouth for offering me a quality research environment. I am also grateful to my fellow students for their encouragement and friendship. I would like to express my appreciation to all my friends and members of the Protégé tool at Stanford University, especially Michael and Alexander for their help, support and valuable insights throughout the development of the ontology model. I am grateful to my colleagues and other researchers who have contributed to this thesis, both indirectly and directly. I warmly thank my colleagues, especially doctoral students in the School of Engineering, for their help and support (Haider, Murtadha, Maji, Sikriu, Ali, Michael, Mahumod, Pricilla, Chenedo, Ahmed and others). I gratefully acknowledge the funding sources that made my PhD work possible as I am sponsored by the Higher Committee for Education Development in Iraq (HCED), Iraq.

I would like to express my deep gratitude to my parents, brother and sisters for their constant support and encouragement and, more importantly, for their faith in me over the course of many years. My parents, I can never thank you enough for what you have done for me. My great appreciation is also extended to my beloved wife, Zahraa Albayati, who has been with me throughout and has sacrificed so much in helping me to pursue my academic journey. Thank you for your patience and care, my life is full of love and happiness because of you. Finally, I would like to express my love to my beautiful sons, Warth and Siraj, for their patience and understanding over the time that I have spent away from them during my PhD journey.

## **DEDICATION**

To all the orphans and poor people in the world, and especially in my country Iraq to which I belong, I wish with this achievement to be able to draw smiles on their faces and to dry the tears from their eyes.

## **LIST OF PUBLICATION**

### **1. Journals papers**

- ❖ Ibrahim, M.E., Yang, Y., Ndzi, D.L., Yang, G. and Al-Maliki, M., 2019. Ontology-based personalized course recommendation framework. *IEEE Access*, 7, pp.5180-5199. DOI: 10.1109/ACCESS.2018.2889635
- ❖ Ibrahim, M.E., Yang, Y. (2019). Automatic ontology construction for information integration in higher education, *Journal of information sciences* (**Under review**)

### **2. Conferences papers**

- ❖ Ibrahim, M.E., Yang, Y. and Ndzi, D., 2017, July. Using ontology for personalised course recommendation applications. In *International Conference on Computational Science and Its Applications* (pp. 426-438). Springer, Cham.
- ❖ Ibrahim, M.E. and Yang, Y., 2019, January. An ontology-based web crawling approach for the retrieval of materials in the educational domain. In *11th International Conference on Agents and Artificial Intelligence*. SciTePress.

## **ABBREVIATIONS**

IR	Information Retrieval
RS	Recommendation System
CF	Collaborative Filtering
CBF	Content Based Filtering
IE	Information Extraction
KNN	K-Nearest Neighbour
OWL	Ontology Web Language
OPCR	Ontology based Personalised Course Recommendation
OM	Ontology Mapping
UCAS	Universities and Colleges Admission Service
DB	Database
HTML	Hypertext Markup Language
TF	Term Frequency
IDF	Inverse Document Frequency
VSM	Vector Space Model
HE	Higher Education
NSS	National Student Survey
HECoS	Higher Education Classification of Subjects
DOM	Document Object Model

# TABLE OF CONTENTS

LIST OF PUBLICATION .....	V
ABBREVIATIONS .....	VI
LIST OF FIGURES .....	X
LIST OF TABLES .....	XII
CHAPTER 1 INTRODUCTION .....	1
1.1    Motivation.....	1
1.2    Aims and objectives.....	4
1.3    Major contributions.....	5
1.4    Thesis organisation .....	6
CHAPTER 2 BACKGROUND AND RELATED WORK .....	8
2.1    Recommender Systems: main approaches and challenges .....	8
2.1.1. Collaborative filtering system (CF) .....	11
2.1.2. Content-Based Filtering Models .....	13
2.1.3 Knowledge-Based Filtering Models .....	17
2.1.4 Hybrid Based Filtering Models.....	18
2.2.    Recommendation systems in the higher education domain .....	20
2.2.1    Course Recommender System .....	23
2.2.3 Challenges in the course recommendation system.....	28
2.3.    Ontology .....	29
2.3.1 Ontology Representation.....	31
2.3.2    Data Extraction Using Ontology .....	32
2.3.3    Ontology Mapping .....	33
2.4    Summary .....	34
CHAPTER 3 OPCR: ONTOLOGY-BASED PERSONALISED COURSE RECOMMENDATION FRAMEWORK.....	36
3.1    Introduction.....	36
3.2    Framework Architecture Design.....	38
3.3    Main Components .....	40
3.3.1    Data Gathering .....	41
3.3.1.1    Course Crawler Module .....	44
3.3.1.2    Job Crawler Module.....	46
3.3.1.3    NSS Score Data Collector.....	47
3.3.1.4    University Rank Collector .....	47

3.3.1.5	Feature Extraction.....	47
3.3.1.6	User Profile data collector (UPDC) .....	48
3.3.2	Ontology Model .....	48
3.3.2.1	Dynamic ontology construction.....	50
3.3.2.2	Ontology Mapping Module.....	52
3.3.3	Recommendation Component.....	53
3.3.3.1	Recommendation Engine .....	53
3.4	User Interface Component .....	54
3.5	Summary .....	55
<b>CHAPTER 4 ONTLOGY MODEL AND RECOMMENDATION ENGINE ALGORITHMS</b>	.....	<b>56</b>
4.1	Introduction.....	56
4.2	Ontology model .....	57
4.2.1	Ontology Construction Module .....	58
4.2.2	Course Ontology .....	60
4.2.3	Student Ontology .....	64
4.2.4	Job Ontology .....	66
4.3	Ontology Mapping .....	68
4.4	Recommendation algorithms .....	71
4.4.1	Content Based Recommendation .....	72
4.4.1.1	Items representation design.....	73
4.4.2	Collaborative Based Recommendation .....	74
4.4.2.1	User Similarity Calculation.....	74
4.4.2.2	Ontology based K-nearest neighbour algorithm .....	78
4.4.2.3	Final Scoring Algorithm .....	81
4.5	Summary .....	82
<b>CHAPTER 5 IMPLEMENTATION AND RESULTS</b>	.....	<b>83</b>
5.1	Data Source and Configuration.....	83
5.1.1	Data Collection Module .....	84
5.1.2	Web Crawler .....	84
5.2	Application of OPCR .....	87
5.2.1	OPCR Interface .....	87
5.2.2	Ontology model.....	91
5.2.2.1	Ontology Construction Module .....	92
5.2.2.2	Ontology Mapping Implementation.....	94
5.2.3	Recommendation Producing .....	96
5.2.3.1	Content Based Filtering .....	98

5.2.3.2	Collaborative Based Filtering .....	99
5.2.3.3	Final Recommendation List .....	100
5.3	Summary .....	101
	<b>CHAPTER 6 EXPERIMENTAL EVALUATION .....</b>	<b>102</b>
6.1	Experimental Study.....	102
6.2	Description of the experiment.....	102
6.3	Data Source and Configuration.....	105
6.4	Evaluation Metrics .....	105
6.4.1	Offline Evaluation.....	105
6.4.2	Online evaluation .....	106
6.4.2.1	Recovery .....	106
6.4.2.2	Accuracy of List Relevance .....	107
6.4.2.3	Rank Accuracy .....	109
6.5	Experimental Results .....	111
6.6	Summary .....	120
	<b>CHAPTER 7 CONCLUSION AND FUTURE WORK .....</b>	<b>121</b>
7.1	Conclusion .....	121
7.2	Limitations .....	123
7.3	Future work .....	124
	<b>REFERENCES .....</b>	<b>125</b>
	<b>APPENDICES .....</b>	<b>139</b>
	Appendix A Course subject classifications.....	139
	Appendix B Similarity calculation function process in CBF and CF .....	141
	Appendix C User and admin Interface pages.....	145
	Appendix D OPCR experiment evaluation form .....	149

## LIST OF FIGURES

Figure 2. 1 Collaborative filtering recommendation.....	11
Figure 2.2 Content-based recommendation .....	14
Figure 2.3 Hybrid systems .....	19
Figure 3.1 OPCR main architecture.....	40
Figure 3.2 Proposed crawler main architecture .....	43
Figure 3.3 HTML structure of a course webpage .....	45
Figure 3.4 HTML structure of a job webpage. ....	47
Figure 3.5 Dynamic Ontology Construction.....	51
Figure 4.1 Data flow and main related module in ontology model .....	58
Figure 4.2 Construct course ontology from course table .....	60
Figure 4.3 Graphical representation of the course ontology in the protégé environment.....	63
Figure 4.4 Course ontology structure.....	62
Figure 4.5 Graphical Representation of the Student Ontology.....	65
Figure 4.6 Student ontology structure.....	66
Figure 4.7 Job ontology structure .....	67
Figure 4.8 Graphical representation of the job ontology .....	67
Figure 4.9 Concepts hierarchy for mapped ontologies .....	70
Figure 4.10 Hierarchical matching and matching parameters .....	75
Figure 5.1 Web crawler interface.....	85
Figure 5.2 Performance comparison ontology crawler and traditional crawler.....	86
Figure 5.3 Performance comparison between ontology crawler and traditional crawler.....	87
Figure 5.4 Interface data flow diagram.....	88
Figure 6.1 Course and job recommendations .....	104
Figure 6.2 Comparison between OPCR and (CBF, CF) performance metrics .....	112
Figure 6.3. Comparison between POCR and UCAS performance metrics .....	113
Figure 6.4. Attitudes Questions .....	114
Figure 6.5. Accuracy Questions.....	115
Figure 6.6. Familiarity Question.....	115
Figure 6.7. Novelty Questions .....	116
Figure 6.8. Diversity Questions .....	117

Figure 6.9. Interaction Adequacy Questions.....	117
Figure 6.10. Perceived Ease of Use Questions .....	118
Figure 6.11. Perceived Usefulness Questions.....	119
Figure 6.12. Control Transparency Questions .....	120

## **LIST OF TABLES**

Table 3.1 Example result of the web crawler for course information .....	46
Table 3.2 Example result of the Web Crawler for job information .....	46
Table 4.1 Factors and keys constituent elements for selecting university courses.....	60
Table 4.2 The main classes of student, course and job ontologies with their relation properties .....	69
Table 4.3 Example of $U_a, U_b$ recommendations history.....	77
Table 4.4 Sample of top k-nearest neighbour courses and their rates .....	79
Table 4.5 Removed courses and their rate if rate $\geq 3$ .....	79
Table 6.1 number of participants and level of study.....	103
Table 6.2 Example of Recovery Metric .....	107
Table 6.3 Example of Accuracy of list relevance .....	108
Table 6.4 Example of the percentage of relevant items to user u .....	109
Table 6.5 Example of Represent System Ranking and User Ranking case1 .....	110
Table 6.6 Example of Represent System Ranking and User Ranking case2.....	111

# CHAPTER 1 INTRODUCTION

*“Education is the most powerful weapon which you can use to change the world”*

Nelson Mandela

This chapter discusses the motivations which demonstrate how timely this research is with regard to the problem it seeks to solve as well as the aims and objectives that the research intends to achieve. Subsequently, the contributions of this thesis in terms of both knowledge and technical perspectives are highlighted. Finally, the organisation of the contents of this thesis is explained.

## 1.1 Motivation

Theoretically, the ever-growing volume of digital content should increase the opportunity to discover content that matches personal needs. However, a user of a conventional information system may experience information overload since only a few of the items are within the field of interest of the user (Bollen, Knijnenburg, Willemsen, & Graus, 2010).

Research shows that students become overloaded by the large amount of information available when choosing a course (Huang, Chen, & Chen, 2013). Information overload occurs when a large amount of information beyond one's capacity to process is communicated. The utilisation of advanced features of educational technologies has provided access to a more productive and more complex information environment in a diversity of formats and from different types of information resources. This propagation of information has imposed information overload on students (Kalyuga, 2011). Nowadays, the range of course-related information available to students continues to rapidly increase (Bhumichitr, Channarukul, Saejiem, Jiamthaphaksin, & Nongpong, 2017). Finding course related information from a large number of websites is a challenging and time-consuming process. An effective search will include all the relevant information about course content, the educational institution and career information regarding a specific course subject. Helping students to make the correct choice from a myriad of available courses in order to meet their individual needs is a real challenge (Huang, Zhan, Zhang, & Yang, 2017).

Such abundant information means that students need to search, organise and use the resources that can enable them to match their individual goals, interests and current level of knowledge. This can be a time-consuming process as it involves accessing each platform, searching for available courses, carefully reading every course syllabus and then choosing that which is most

appropriate for the student (Apaza, Cervantes, Quispe, & Luna, 2014). This abundance of information has created the need to help students to choose, organise and use resources that match their objectives, interests and present knowledge (Farzan & Brusilovsky, 2006). Bendakir and Aïmeur report that students pursuing education are faced with two challenges: a myriad of courses from which to choose and a lack of knowledge about which courses to follow and in what sequence(Bendakir & Aimeur, 2006).

The process of choosing a course can be incredibly tedious and extremely complicated. Nowadays, students can rapidly find information relating to universities and the courses offered by them using online resources (Huang et al., 2013). However, simply because more course information is now available from university websites does not automatically mean that students possess the cognitive ability to evaluate them all (Ibrahim, Yang, & Ndzi, 2017). Instead, they are confronted with a problem that is termed “information overloading” (Z. Zhang, Zhou, & Zhang, 2010).

Artificial intelligence methods developed at the beginning of research are now being applied to information retrieval systems. Recommender systems (RS) provide a promising approach to information filtering (Garcia, Sebastia, & Onaindia, 2011) as they help users to find the most appropriate items (Jannach, Zanker, Felfering, & Friedrich, 2011). There are many online systems currently available that can be used to find and search for courses. However, none of these are sufficiently targeted to provide the user with personalised recommendations which offer comprehensive information about specifically relevant courses.

One of the motivations for this research is to reduce the information overloading that users face when they wish to select a university course. Education information is published on the internet in different formats and therefore extracting useful information that meets with the user’s search query presents a significant challenge (Alimam & Seghiouer, 2013). The heterogeneity of course information and personal user needs makes the decision process very tedious and complicated. Measuring the ontology hierarchy structure of item concepts is one of the promising methods which can help to tackle the heterogeneity problem (Bach & Dieng-Kuntz, 2005).

Furthermore, although some course titles are similar, each can lead to a different career path (DS & K, 2015). Sandvig and Burke argued in their research work that a lack of knowledge regarding which appropriate item to choose from a large number of items means that people need to seek an advisor or guidance(Sandvig & Burke, 2005). Providing comprehensive

knowledge about a satisfactory item that a user may wish to select is another challenge because the difference in a user's tastes and preferences will influence the degree of user satisfaction. For example, a person seeking to choose a university course degree will need to acquire relevant information regarding the course, not simply the course subject content, but also the reputation of the university, the facilities provided, career opportunities and so forth. Therefore, the need to establish a comprehensive framework that can extract and integrate information from multiple sources and align this data in a unified form is another motivation for this research.

Recommender systems offer a promising approach to information filtering (Garcia et al., 2011) as they help users to find the most appropriate items (Jannach et al., 2011). Based on the needs of each user, the recommendation system will generate a series of specific suggestions (Ren, Zhang, Cui, Deng, & Shi, 2015). Recommender systems have been used to provide recommendations in a variety of domains such as e-commerce, news, movies, music, research papers, course materials among others. The education domain has used recommender systems for different purposes such as e-learning applications, academic advice, course material suggestions and so forth. Many online systems are currently available that can be used to find and search for courses (S. Wang & Sapporo, 2006) which use tools based on the users' prior knowledge of the courses (H. Zhang, Yang, Huang, & Zhan, 2017), keyword-based queries (Khan, 2000; ucas.com, 2018), collaborative filtering (CF) (Carballo, 2014) (T. Huang et al., 2017), data mining and association rules (Noakes, Arrott, & Haakana, 1968; H. Zhang et al., 2017) and content-based filtering (CBF) models (Lotfy & Salama, 2014). Despite the strong influence of existing course recommendation systems and how useful they can be, there are certain significant limitations such as:

- Models based mainly on keywords fail to address an individual user's needs in the recommendation process.
- Although models use collaborative filtering and data mining such as association rules and decision trees, there is often a lack of historical information that makes this approach challenging to adopt. For instance, new students who wish to use the system do not have sufficient information about the model and are therefore unable to generate any recommendations.
- The shortcoming of models that use content-based filtering is that current approaches are based only on a specific subject recommendation rather than an entire university course. Moreover, the similarity calculation in these models is based on the weighted average of

features and does not take into account user interaction with the system, such as the rating value of recommended items.

- Another shortcoming of current models is that they do not provide comprehensive knowledge regarding the course that is most relevant to the student. For example, students need to know what future career the course will lead to and require information regarding this aspect, as well as the quality of the facilities of the educational institution itself that will be providing the course.

Categorising the needs of students and their areas of interest enables an appropriate course to be recommended. It is possible to help students to select a course by developing methods that will both integrate the data from multiple heterogeneous data sources and allow this to rapidly establish valuable course-related information (Huang et al., 2013).

All these facts provided the motivation to develop a new approach to overcome the information overloading phenomenon and to obtain comprehensive knowledge regarding the recommended items. Two research problems need to be addressed. First, how to integrate all available information about courses, including the course modules, job opportunities and the users' interests and build a relationship between the relevant information. Second, with all the integrated information, how to recommend the most relevant courses to meet user's individual needs.

## **1.2 Aims and objectives**

This thesis aims to tackle the problem of information overloading. It develops a practical framework based on the methods proposed in the research that can have a realistic application with an impact within the scope of an education recommender system. The framework supports data integration and course recommendation applications. Involving algorithms enables intelligent course recommendations to be produced based on data integrated from multiple sources. The ultimate aim is the ability to provide a personalised recommendation from a wide range of data sources, focusing on a student's individual needs when choosing a course.

The aim of the ~~thesis~~ research can be divided into the following specific objectives:

1. To study the state of the art of recommender systems, particularly focusing on those that have been applied in the education domain. Also to examine the tools available to students for assistance in decision making when choosing a suitable course to meet their personal needs.

2. To study the tools of preference modelling, concentrating on the methods that employ user profiles, and primarily analyse how they deal with the problems of initialisation and dynamic updating of the profile. To design a model to dynamically manage user and item profiles that provides improvement to the performance of conventional recommender systems.
3. To develop methods to integrate data from multiple heterogeneous data sources which will allow a user to quickly access valuable course-related information based on the user's preferences thereby assisting the user to choose course relevant to their career direction.
4. To develop a framework that can be used by perspective students who plan to choose university courses. The framework should be able to provide personalised recommendations that meet with the individual student's needs by combining different types of recommendation techniques. It should be able to support automatic data extraction, integration and personalised course recommendations.
5. Based on the framework, to design and implement a personalised course recommendation system to demonstrate the feasibility of the proposed approaches in a real application.

### **1.3 Major contributions**

This research addresses the existing gap and investigates an approach by which to automatically extract and integrate course information based on ontology technology and to enhance the performance of a recommendation system by reducing information overloading in the education domain. The aggregation of ontology domain knowledge into the recommendation process is one of the solutions that can overcome the limitations of conventional recommender systems. Ontology-based (OB) recommenders systems are knowledge-based and use ontology to represent knowledge about the items and the users in the recommendation process. In addition, user profiling that is based on ontology, item ontology and the semantic similarity between two ontologies is used to overcome the new user problem.

The main contributions of this thesis lie in the following points:

1. It contributes to the knowledge of current recommender systems by adding insight as to how existing problems are usually tackled and why there still remain shortcomings.

2. It defines a novel Ontology based Personalised Course Recommendation (OPCR) framework by combining several artificial intelligence techniques including collaborative filtering, content-based recommendations, ontological representation and management of knowledge. A set of ontology based recommendation algorithms are developed for personalised recommendation. The framework is thus capable of automatic data extraction, integration and personalised course recommendations to provide students with suitable recommendations to meet with their needs. It aims to not only increase the precision metrics but also to reduce information overloading.
3. The ontology model, designed to extract and integrate information from multiple sources, contributes to improving the diversity of recommendations by overcoming the heterogeneity of course information. In addition, it features properties such as generality which enable it to be used in different recommendation system domains which change with the user's interests and the object's attributes.
4. A personalised recommendation system is developed and evaluated. The system is available online as open access for users.

## **1.4 Thesis organisation**

This thesis is organised as follows. Chapter 1 Introduction presents a contextualisation of the work and offers a brief explanation of its motivations. The general concepts are clarified and a description of the contributions of the thesis is provided.

Chapter 2 discusses background details and related work and research on recommendation systems and the aspect of ontology. It also highlights different recommendation algorithms and the main challenges faced by general recommender systems, particularly in the education domain. Attention is mainly focused on the collaborative and content-based systems with the strengths and weaknesses of each model being pointed out and an analysis of the research trends in the area of recommender systems is provided.

Chapter 3 presents the Ontology based Personalised Course Recommendation (OPCR) framework and its components in great detail, constituting the core of this work.

Chapter 4 expands the proposed ontology model and its modules and also discusses in detail the recommendation filtering algorithms that are used within the framework.

Chapter 5 continues with the implementation of the actual recommender system, namely OPCR, and its intermediate steps until the generation of recommendations followed by the results of this implementation.

Chapter 6 discusses details of the different approaches of OPCR evaluation followed by the results of user satisfaction measurements.

Chapter 7 concludes by summarising the goals of the thesis, defines the contributions provided and ends with an analysis of future directions for this research.

Finally, the thesis includes the bibliographic references used for its elaboration and 4 annexes that provide information relevant to the thesis.

## CHAPTER 2 BACKGROUND AND RELATED WORK

*“Research is to see what everybody else has seen, and to think what nobody else has thought.”*

Albert Szent

This chapter discusses the background to the thesis topic and related work regarding the relevant literature. It provides an up-to-date, state of the art solution in the field of intelligent recommender systems (RS) within the education domain that may be useful, not only to scientists working within this field but to designers and developers of intelligent recommender systems in other domains.

It also discusses how this research relates to previous works undertaken in this area and in what ways it significantly differs from these. Section 2.1 aims to present the theory concerning the recommendation system approaches, a set of features for recommender systems is defined, and a review of the current state of the art is conducted together with an explanation as to why current solutions are not sufficient to address the problem of information overloading.

Section 2.2 refers to recommender systems in a specific field that in this thesis, is course recommendations in higher education, reviews the main shortcomings of current solutions in this area and explains how the proposed system addresses the information overloading problem in the field. Section 2.3 discusses the aspects of ontology in recommendation systems, particularly in the education domain, and how using ontology can extract and integrate information from multiple sources for utilisation in a unified form in order to enhance both the performance of the recommender system and user satisfaction. Moreover, related concepts such as ontology construction, ontology mapping and main challenges are discussed. Finally, section 2.4 presents a summary of the chapter.

### 2.1 Recommender Systems: main approaches and challenges

The rapidly increasing scope of the internet has given users the facility to choose from an enormous range of information, whether this is information concerning their education, experiences in their world or information that enables them to maintain their lifestyle.

Essentially, to offer a straightforward description, a recommender system can provide recommendations (suggestions) to users in different contexts, such as when they have to choose between a large numbers of items or when they wish to receive suggestions. Recommender systems become particularly helpful in situations where there is an *information overload*

*problem*, that is, the remarkable array of choices makes the search and selection a challenging task for the user. Information overload was a term introduced to represent the feeling of exhaustion and confusion that occurs because of the cognitive energy required to manage the number of information users has to deal with.

Recommender systems produce a set of technologies and algorithms from various fields such as information retrieval, machine learning, marketing, education, economics and many others. It has become popular since the mid-1990s, contributing solutions to the problem of information overload on the World Wide Web. Different approaches have been manipulated, each with their advantages and shortcomings. Given the fact that recommender systems are generally established to solve real-world problems, the field is exciting and fulfilling to both the academic domain and business world.

Resnick and Varian (Kembellec, Chartron, & Saleh, 2014) define a recommender system as “a system able to learn users’ preferences about different items and use these preferences to propose new items that users might be interested in”. Burke (Burke, 2002) adds a new notion regarding the definition of recommender systems, “a recommender system must be able to provide individualised recommendations and guide users in a personalised way”. Recommender systems began to attract consideration from researchers in the early nineties (Goldberg, Nichols, Oki, & Terry, 1992). Research into recommender systems spread beyond information retrieval and filtering analysis and began to be applied to a variety of different domains. The object of using recommender systems is to overcome information overload by retrieving the most appropriate information and services from a massive amount of data.

Recommender systems are used by many e-Commerce websites, such as Amazon, to help customers to find appropriate products (Linden, Smith, & York, 2003). In recommender systems, the items can be recommended based on specific information which can be acquired from the demographic data of customers, an analysis of the past purchasing behaviour of consumers as a prediction for future buying behaviour or from the top overall sellers (Adomavicius et al., 2011). However, their application has been extended to fields such as movies, music suggestions, news, bookstores and education (Al-Badarenah & Alsakran, 2016; Cantador, Bellogin, & Castells, 2008; Cui & Chen, 2009; Hsu, 2008; Jones & Pu, 2009; D. H. Park, Kim, Choi, & Kim, 2012). The main aim of using recommender systems is to reduce information overload by the retrieval of the most relevant information and services from a vast amount of data.

Furthermore, recommender systems can be defined primarily as software programmes that attempt to recommend items to users by predicting users' item preferences based on various types of information, including information about the items, the users and the interactions between users and items. The performance of recommendation systems is influenced by many factors that can affect the quality of recommendations according to the application domain. In their work, Martinez and Lhadj (Martinez & Lhadj, 2013) highlighted the main factors that can influence the results of a recommendation system as shown in Table 2.1.

<b>Factor</b>	<b>Description</b>
User factor	This includes all basic characteristics such as background, demographics and language
Personal factor	Behaviour, flexibility to accept or reject recommendations, interest, mood, motivation, trust, intuition and honesty, privacy, awareness of other options, bookmarks, needs, interaction weight, interaction preferences, interactions between users
Recommendation	Quality, credibility, measurability and weight of the recommendation, reliability, classification, date and time. It is also recommended to include an explanation about why a resource is recommended and who the contributors are.
Resources	Content, thesaurus, taxonomy, tags, keywords, ratings, reviews, summary, contributors, date, number of votes
System	Accessibility, usability, parameters, goal, initial data, data analysis techniques, design, architecture, graphical interface

Table 2. 1 Factors that influence the recommendation for recommender systems

The widely utilised recommendation system filtering techniques can be categorised into four main approaches. The content-based filtering (CBF) approach recommends items similar to those preferred in the past by the user (Lops, de Gemmis, & Semeraro, 2011). The collaborative filtering (CF) approach recommends items preferred by users with similar needs or interests (Kim, 2013). Knowledge-based filtering recommends items whose features meet users' needs and preferences based on particular domain knowledge (Ruotsalo & Hyvönen, 2007). A hybrid recommendation system is an approach that combines two or more recommendation techniques to overcome the typical shortcomings of each approach (Adomavicius & Tuzhilin, 2005a; Burke, 2002; Ibrahim, Yang, Ndzi, Yang, & Almaliki, 2018). In the following subsections, these techniques are described in detail, and their respective advantages and shortcomings are studied.

### 2.1.1. Collaborative filtering system (CF)

Collaborative filtering is one of the broadly used techniques in recommender systems in order to overcome the information overloading problem. The idea behind this technique is to assist people in making their own decisions based on the opinions of other people who share similar interests (Kaminskas & Bridge, 2014). A large community of users is required in order to be able to collect and analyse an immense amount of information regarding user behaviour and characteristics as shown in Fig 2.1. Collaborative filtering systems suffer from a *cold start problem* when there is a lack of data regarding a current user (Adibi & Ladani, 2013). To tackle this problem, the system can offer the top rated item.

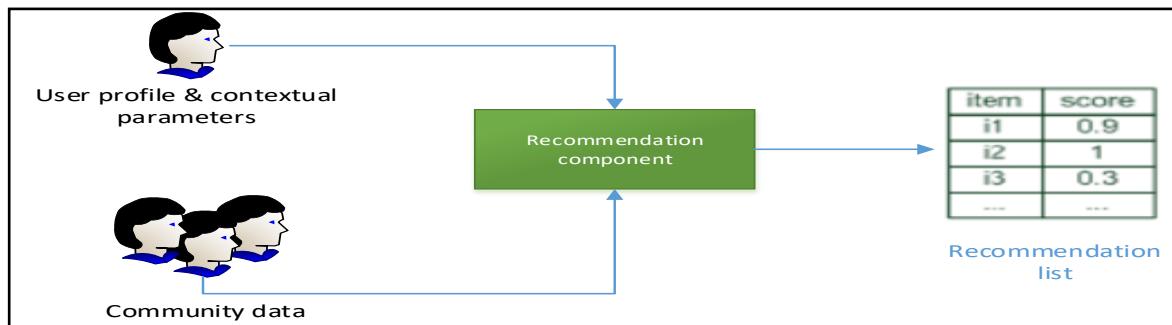


Figure 2. 1 Collaborative filtering recommendation

Collaborative filtering is considered to be the most popular and widely implemented technique in recommender systems. Since 1990, numerous recommender systems based on the collaborative filtering technique have been created and developed in the worlds of academia and business. These systems have been utilised in many disciplines and uses include suggesting courses, news, articles, movies, products, books, web pages and so forth (Cui & Chen, 2009; Herlocker, Konstan, & Riedl, 2000; Linden et al., 2003; Ray & Sharma, 2011; Ren et al., 2015).

According to previous researchers, there is a number of collaborative filtering algorithms that can be applied to generate recommendations. Collaborative filtering algorithms are mainly divided into two classifications; *memory based* and *model-based* algorithms (Bagherifard, Rahmani, Nilashi, & Rafe, 2017).

**Memory-based** collaborative filtering utilises user-item rating data to measure the similarity between users or items to provide recommendations (Zhao & Shang, 2010). It is widely used in commercial systems. The fact that the similarity between users is computed by utilising only rating data means that the system is flexible for any products. Nevertheless, the main flaw in this technique is that, since it is calculated using only rating data, the similarity cannot take into

account the causes that led to a good or bad rating. Therefore, two users might have liked the same item but for certain different reasons. In Chapter 4 of this thesis, a new method is presented that allows a recommender system to include this new dimension. The popular memory-based method is called the neighbour method that is divided into user-based and item-based. In the user-based method, the similarity between users in their consumption models is used to calculate recommendations. For a target user, the preferences of similar users and the neighbours can assist in recommendations (Desrosiers & Karypis, 2011).

In the neighbour formation case, the similarity between the target user and all other users has to be computed. Several algorithms provide a measure for user similarities such as Pearson's correlation (Tang & McCalla, 2009a) and the cosine-based approach (Chang, Lin, & Chen, 2016) which will be explained next.

$$sim(u_i, u_j) = \frac{\sum_{b_k \in K_{ij}} r_i(b_m) (r_i(b_k) - \bar{r}_i) (r_j(b_m) - \bar{r}_j)}{\sqrt{\sum_{b_k \in K_{ij}} (r_i(b_m) - \bar{r}_i)^2} \cdot \sqrt{\sum_{b_k \in K_{ij}} (r_j(b_m) - \bar{r}_j)^2}} \quad (2.1)$$

Pearson's correlation coefficient is a measure for the strength and direction of a linear correlation between two variables. Eq.2.1 presents the Pearson correlation coefficient that computes the similarity between two users. In this equation,  $K_{ij}$  refers to the set of items that are rated by both users  $u_i$  and  $u_j$ .  $r_i(b_m)$  is  $u_i$ 's rating from item  $b_m$  and  $\bar{r}_i$  is the average rating value for user  $u_i$ .

In the cosine-based approach, the users are admitted as vectors  $u_i$  and  $u_j$  in a  $m$ -dimensional space, where  $m = |K_{ij}|$ . The vectors thus represent the rating values for the items that were rated by both users. The similarity is then computed as the angle between those two vectors, as shown in Eq.2.2.

$$sim(u_i, u_j) = \cos(\vec{a}_i, \vec{a}_j) \quad (2.2)$$

$$sim(u_i, u_j) = \frac{\vec{a}_i \cdot \vec{a}_j}{|\vec{a}_i| \cdot |\vec{a}_j|}$$

$$sim(u_i, u_j) = \frac{\sum_{b_k \in B_{ij}} r_i(b_k) \cdot r_j(b_k)}{\sqrt{\sum_{b_k \in B_{ij}} r_i(b_k)^2} \cdot \sqrt{\sum_{b_k \in B_{ij}} r_j(b_k)^2}}$$

In the item-based approach, the similarity between items with familiar users is employed. The idea behind this is that items that are similar to those that the user has previously rated or utilised are good candidates for a recommendation (Sarwar, Karypis, Konstan, & Riedl, 2001).

**Model-based** collaborative filtering uses the user-item rating data to build a model that is then used to make predictions for unrated items. The process of building a model can be performed using different statistical and machine learning algorithms (Kumar, 2011). Machine learning techniques inspire these methods, for example, Bayesian networks, artificial neural networks (Salakhutdinov, Mnih, & Hinton, 2007) and latent factor models (Koren, Bell, & Volinsky, 2009).

### Cold start problem

These systems face a cold start problem (Schein, Popescul, Ungar, & Pennock, 2002) when there is no available data regarding a current user. The cold start problem occurs when the recommender system cannot draw any assumptions for users or items about which it has not yet collected sufficient information. The first time a user visits a specific recommender system, for instance, none of the items will have been rated. Therefore, the recommender system does not know what the likes and dislikes of that user are. The same problem occurs when a new item is added to the recommender system. Since nobody has ever rated that item, the recommender system cannot know to which other items it is similar. Consequently, the recommender system cannot recommend the item until a large number of users have rated it. However, one of the advantages of collaborative filtering is that it does not need or require any knowledge about the items in the database, the matrix of users' ratings is the only input necessary.

#### 2.1.2. Content-Based Filtering Models

Content-based filtering is a conventional method which is applied when information overload problems need to be dealt with (Pazzani & Billsus, 2007a). This filtering technique recommends items for the user based on information regarding previously evaluated items for that user. However, this technique suffers from over-specialisation, as it is incapable of determining unexpected items and the user will only receive recommendations for items similar to those that the user has rated before. This problem of novelty is also known as the serendipity problem.

Unlike the collaborative filtering approach, the content-based filtering approach uses content that the user has liked in the past in order to suggest similar content and, as opposed to collaborative filtering, does not utilise other users' interests to issue recommendations (Burke, 2002; Pazzani & Billsus, 2007b). This approach analyses the information from items that have been previously rated. The process of recommendation essentially involves locating suitable matches between the user profile and the characteristics of the items. One of the advantages of content-based methods is that they can deal efficiently with the new item problem, that is, they can recommend new items for which there is no user feedback, unlike collaborative filtering algorithms.

Furthermore, content-based filtering has proved popular for producing recommendations for information items when, for example, the user marks/buys certain items of interest and the system then offers the items which are most similar items to the user's favourite items. These systems need a great deal of detail about the items in the database to be able to recommend similar ones and about the user profile that describes what the user likes. On the other hand, it does not require a large community of users as Fig. 2.2 shows. Every content-based recommender system has the following three goals:

- to analyse item descriptions and documents
- to build a user profile
- to compare favourite items with other items in the database

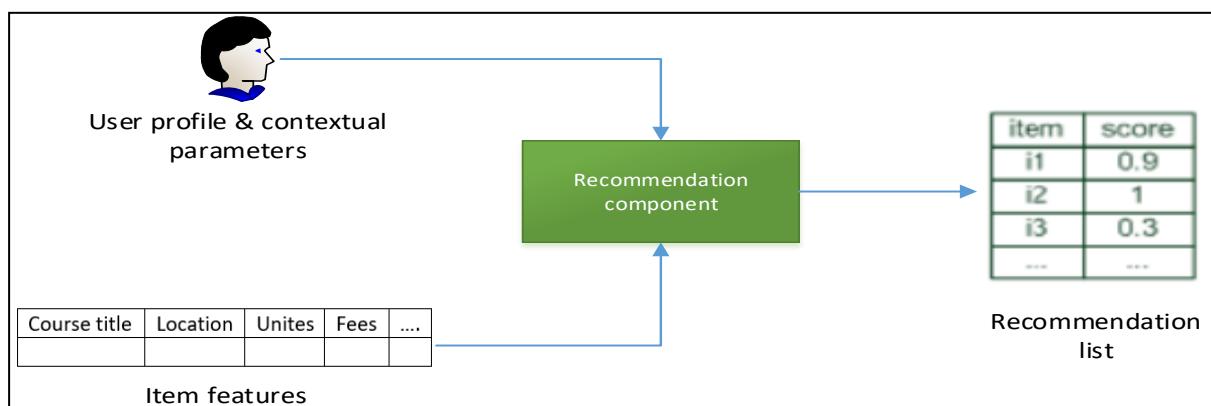


Figure 2.2 Content-based recommendation

The architecture of such a recommender system was published in (Lops et al., 2011). The authors describe the three main components that contribute to reaching the above goals:

- *The content analyser* provides extraction of structured, relevant information (usually keywords) from texts to be able to process these further.

- *The profile learner* gathers information about user interests such as what item they select, rate or leave other feedback for (Ruthven & Lalmas, 2003), employs machine learning algorithms (Han, Kamber, & Pei, 2012) and creates a user profile.
- *The filtering component* utilises the user profile to recommend relevant items by matching the user profile to corresponding items in the database. It uses different similarity metrics (Zezula, Dohnal, & Amato, 2006), cosine similarity, being one of the most often used as presented in Eq.2.2, creates a ranked list of items and suggests these to the user.

Proposals for content-based recommendation algorithms attract perspectives and algorithms from different domains such as information retrieval, semantic web and machine learning. For instance, term-weighting models from information retrieval were employed in early proposals for web recommendations (Balabanović & Shoham, 1997). However, in the current content-based filtering approaches, the data is not modelled correctly. Using ontologies to model the data provides a better modelling quality and thus more suitable recommendations because better-modelled items mean more rigorous user preferences have modelling abilities which can produce more accurate similarities (Madel, Shoval, Shapira, & Taieb-Maimon, 2010). Methods using semantic web technologies have also been introduced for content-based recommendations, as in the case of news recommendation (Cantador et al., 2008; Kumar, 2011), or movie and music recommendations leveraging Linked Open Data (Ostuni, Di Noia, Mirizzi, & Di Sciascio, 2014). Regarding the use of machine learning techniques, Mooney and Roy (Mooney & Roy, 1999) used Bayesian classifiers for book recommendations, and Pazzani and Billsus (Pazzani & Billsus, 1997) used numerous techniques such as Bayesian classifiers, clustering, decision trees and artificial neural networks for website recommendation.

### **Limitations of the content-based approach**

Content-based recommender systems have several limitations which have been identified in the literature (Burke, 2002; Ekstrand, Harper, Willemsen, & Konstan, 2014). The most relevant of these are:

- Limited content analysis. Content-based recommendations are constrained by the features that are explicitly associated with the items to be recommended. For example, content-based course recommendations can only be based on material written about a course: course title, course fee, university name. The effectiveness of these techniques thus depends on the available descriptive data. Therefore, in order to have a sufficient set of features, the content should be either in a form that can be automatically parsed by a

computer or in a form in which the features can be manually extracted easily. In many cases, these requirements are very difficult to fulfil. There are certain domains where automatic feature extraction is complicated and to assign features manually is often not practical. For instance, even if a recent attempt underlines the need for further research in this direction (Li, Ogiara, & Li, 2003), it is much harder to apply automatic feature extraction methods to multimedia data such as graphical images, video streams and audio streams, than it is for text content. A recent trend is to enrich content representation by means of external knowledge sources, such as ontology based ones. The Explicit Semantic Analysis (ESA), introduced in (Markovitch & Markovitch, 2006), proposes an indexing technique based on content gathered from Wikipedia articles. An early attempt of coupling content-based filtering based on ontology with techniques for knowledge infusion is proposed in (Musto, 2010).

- Content over-specialisation. Content-based filtering based on ontology retrieves items that score highly against a specific user profile. Content-based techniques cannot recommend items that are different from anything the user has seen before. Thus, for instance, a person with no experience in ambient music will never receive recommendations about that genre if he has never enjoyed something at least similar to it. To overcome such limitations, it may be appropriate to introduce an element of randomness in the recommendations (Maes & Sharadanand, 1995). Alternative approaches, such as that implemented in DailyLearner (Billsus & Pazzani, 2000), propose to filter out items not only if they are too different from user's preferences but also if they are too related to something the user has viewed previously. Furthermore, in (Zhang, Callan, & Minka, 2002) a set of five redundancy measures is provided in order to evaluate whether a document that is deemed to be relevant contains some novel information as well.
- Cold-start. Before a content-based recommender system can really grasp user preferences and provide reliable recommendations, each user has to rate a sufficient number of items. However, the recent explosion of Web 2.0 and social platforms has changed the rules for user profiling since, in principle, it is possible to reuse the information the user has already provided (such as comments, posts, tags or data gathered from social networks) and to exploit such information as a starting point to incrementally build and model the user profile. In this area, a recent trend is represented by social media-based user profiling (Bu et al., 2010).

It is significant to note that content-based techniques are not based on the real content of items. For instance, in the context of course recommendation, a content-based filtering system does not concern itself with the content of the course, it is merely based, at best, on the description, keywords or course title. Most of the time, the “content” is simply the genre, author or other metadata. Also, content-based filtering methods do consider the textual content that has been written about items by users, blogs or whatever. They generally apply semantic analysis by using ontologies.

### **2.1.3 Knowledge-Based Filtering Models**

These techniques are usually utilised for an explicit representation of knowledge, as with a case-based reasoning system, ontology or other forms of rule systems. Items are recommended to users according to inference about a user’s interests (Middleton, De Roure, & Shadbolt, 2009). A case-based system relies on the notion of using past problem-solving experiences as a primary source to solve a new problem.

A good example of this knowledge-based method (KB) is presented in the work of Burke (Burke, 2002). In his work, the model was designed to help a user to find restaurants that matched his/her preferences through the use of interactive dialogue. The user could change the retrieved suggestion by refining the search query based on user interest until achieving the suitable option. The other type of knowledge-based method is the use of semantic similarity to recommend items to users. Ontologies have been applied to a variety of recommender systems to reduce content heterogeneity and to improve content retrieval. For example, in (Obeid, Lahoud, El Khoury, & Champin, 2018), good results to cope with content heterogeneity have been obtained by using subsumption hierarchies to generalise user profiles.

Furthermore, the concept of the semantic web has been used to improve e-learning. In (Yang, Sun, Wang, & Jin, 2010), Yang et al. proposed a semantic recommender system approach for use in e-learning to help learners to define suitable learning objectives. Moreover, the system could assist instructors by suggesting new resources that could be adopted to enhance the syllabus of the course. This system has been built with a query keywords extension and uses both semantic relations and ontology reasoning. The authors in (Ren et al., 2015) presented a personalised ontology-based recommendation system which is similar to the two approaches mentioned above. It represents items and user profiles in order to provide personalised services that use semantic web applications. The evaluation shows that the semantics-based methods of the recommender system improve the accuracy of the recommendations.

A recommendation system based on ontology can also solve the cold start problem which occurs when using information from the past is insufficient (Zhou, Yang, & Zha, 2011). Indeed, this problem occurs due to an initial lack of ratings for new users and hence it becomes impossible to make reliable recommendations. An ontology-based model has been proposed for e-learning personalisation which would recommend learning objectives by judging the past preference history of learners. Like traditional systems, this system suffers from a new user problem and is limited to learning objectives only (Ambikapathy, 2011). Ontology structures significantly improve the ontology which can lead to increased accuracy (Bagherifard et al., 2017). For instance, all of the “IS-A”s relations in the ontology for measuring semantic similarity were considered to be similar in a hierarchical tree in which the associations between the concepts were shown by “IS-A”. Calculating the similarity between the two concepts is made less accurate by this.

The knowledge-based recommender system has certain advantages and disadvantages (Adomavicius & Tuzhilin, 2005b; Sieg, Mobasher, & Burke, 2010). On the positive side, it has no cold start problem; the system can recommend items to a new user based on simple knowledge of his/her preferences. It will therefore not require the user to rate or buy very many items in order to provide satisfactory recommendations. On the negative side, the knowledge-based method system faces a scalability problem where more time and more effort are needed to calculate the similarities for a larger case-base compared to other standard recommendation techniques. The knowledge-based method has a further weakness, which is that the system needs to include some information about items, users and functional knowledge in order to produce recommendations.

#### **2.1.4 Hybrid Based Filtering Models**

Hybrid systems utilise a combination of the methods mentioned above to overcome their disadvantages and utilise their strengths (Burke, 2002). For instance, collaborative filtering techniques fail to handle the new-item problem, i.e. they are unable to recommend items that have not yet been rated. However, new items characteristics (information) are generally available and can be used with content-based methods as shown in Fig. 2.3. The first type of hybridisation is to select different approaches and to allow each to produce a separate ranked list which is then merged into one final list (Claypool et al., 1999). Several other hybrid

approaches are based on one model which uses the second model only to overcome shortcomings, e.g. the cold start problem or to improve user profiles (Pazzani, 1999).

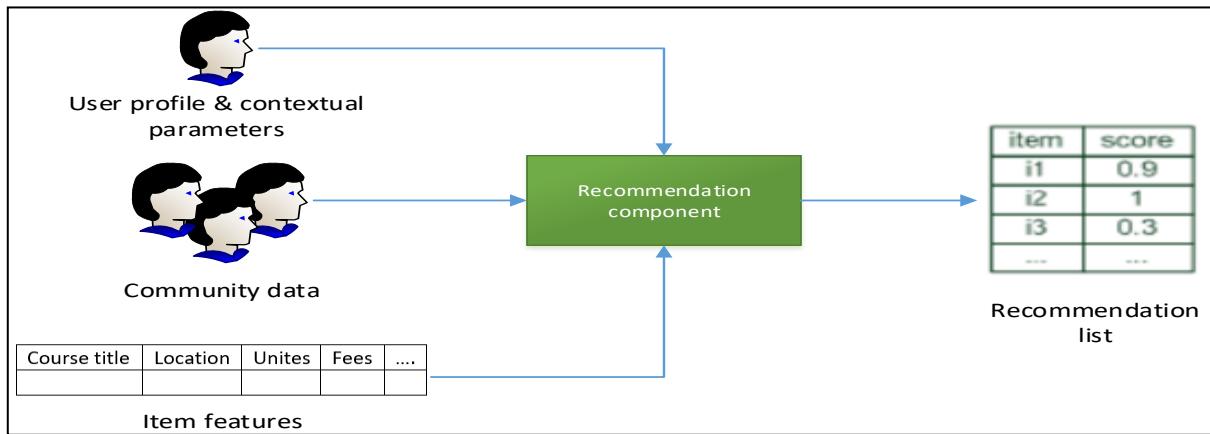


Figure 2.3 Hybrid systems

Burke (Burke, 2002) defined the taxonomy for the hybrid recommendation systems which he listed into the following seven categories:

- **Weighted:** Several recommendation component scores are combined numerically. This category aggregates scores from each factor using an additive formula. For instance, the easiest combined hybrid would be a linear combination of recommendation scores, as in P-Tango (Claypool et al., 1999). This method initially provides equal weight to collaborative and content-based recommenders. However, the weighting is constantly adapted according to users' feedback on predictions.
- **Switching:** From the available recommendation components, the system chooses a particular component and applies the one selected. As an illustration, the recommendation process starts with a content-based recommender, but switches to a collaborative one when the confidence level on the recommendations already presented are not sufficient as shown in the work of Billsus and Pazzani (Billsus & Pazzani, 2000). However, this switching hybrid does not entirely avoid the ramp-up problem, since both the collaborative and content-based systems feature the new user problem.
- **Mixed:** Different recommender systems produce their recommendations that will be introduced together. This method is based on the merging and presentation of multiple rated lists into a single rated list. This method is implemented in the PTV system (Smyth & Cotter, 2000)

- **Feature Combination:** Contributing and actual recommenders are the two different recommendation components that exist for this category. The working of an actual recommender is dependent on the data modified by the contributing recommender which throws features of one source onto the other component's source.
- **Feature Augmentation:** This category is similar to the feature combination hybrid with the only difference being that the contributor gives novel characteristics. It is more flexible than the feature combination method.
- **Cascade:** This category plays the role of tiebreaker. Here, every recommender is assigned a certain priority and, according to that assigned priority, lower priority recommenders play a tiebreaker role over those with higher priority. Usually, one recommendation system is applied to generate a set of candidates and the second algorithm filters and re-ranks this to produce the final list. The cascade hybrid is generally more efficient than a weighted one that applies all of its methods to all items.
- **Meta-level:** The model generated by one of the recommenders is used as the input for another recommender. As stated in (Burke, 2002): “this differs from feature augmentation: in an augmentation hybrid, we use a learned model to generate features for input to a second algorithm; in a meta-level hybrid, the entire model becomes the input.”

## 2.2. Recommendation systems in the higher education domain

*“Applying to university is a big decision, and we want to ensure that all students, whatever their background, have the key facts at their fingertips to help them make the right choice for them.”*

*Dr Vince Cable*

As highlighted in section 2.1, the recommendation problem widely applies to many domains although its main uses are within the fields of e-commerce and entertainment. Most developed systems aim to help the user to decide which products to buy (or consume). Successful application of e-commerce recommenders are reported elsewhere (Sarwar, Karypis, Konstan, & Riedl, 2000).

This success, as mentioned earlier, has motivated the implementation of recommender systems in the educational domain (Manouselis, Vuorikari, & Assche, 2010). In this domain, the ultimate goal is that learners acquire knowledge and that educators support the learning process.

In other words, there are clear differences that impinge on how to design recommender systems for each domain (Drachsler, Hummel, & Koper, 2008a). Recommending items in the education domain, either directly or indirectly, has the goal of improving the decision making process and the selection of appropriate courses. In the education domain, a recommendation system is an intelligent agent that suggests different alternatives to students which takes into account, as a starting point, the previous action from other students with approximately the same characteristics, such as academic performance and other personal information (Park, 2017). It is a fact that, before taking a course, the student has to enrol on the course. However, the most notable aspect of this process is not the enrolment itself, but the decision before that which has to be taken(Ibrahim et al., 2018).

In this section, the most important literature regarding these types of recommendations systems with reference to the dimension of the items recommended and the different approaches that have been used to recommend items in the education domain will be reviewed. However, recommender systems in education are entirely different from recommender systems in e-commerce, as they have to consider not only the students or the educator's preferences for certain learning materials but also how this material may help them to obtain their goals (Bozo, Alarcón, & Iribarra, 2010). In table 2.2, a comparison has been made of the important differences and factors between a general purpose recommendation system and an educational recommendation system.

Difference factor	General recommendation system	Educational recommendation system
The goal	In fields such as e-commerce, a user is looking to buy a product of a specific quality and in a specific price range (Nikos Manouselis, Drachsler, Vuorikari, Hummel, & Koper, 2011)	Educational recommender systems help the user or a group of users to find suitable resources and learning activities for optimum achievement of learning goals and the development of competences in less time (Drachsler et al., 2009; Drachsler, Hummel, & Koper, 2008b)
The context	Most recommender systems share factors such as networks and peer information (J. Hu & Zhang, 2008; Ramadoss & Balasundaram, 2006; Santos & Boticario, 2009; T. Tang & Mccalla, 2004)	The context of educational recommender systems is pedagogically related. Factors that should be considered as part of the context are pre and post-requisites, timeframe and instructional design (Abel, Bittencourt, Henze, Krause, & Vassileva, 2008; Santos &

		Boticario, 2009), and social networks (Q. Yang et al., 2010)
User influenced by	Recommender systems are mostly based on user tastes, personal preferences or what a user likes or dislikes (Santos & Boticario, 2009; T. Y. Tang & McCalla, 2009b)	Highly influenced by pedagogical factors such as learning history, knowledge, preferences, processes, strategies, styles, patterns, activities, feedback, misconceptions, weaknesses, progress and expertise (Abel et al., 2008; Gasparini, Lichnow, Pimenta, & Oliveira, 2009; Masters, Madhyastha, & Shakouri, 2008; Prieto, Menéndez, Segura, & Vidal, 2008; Wan, Ninomiya, & Okamoto, 2008; Q. Yang et al., 2010)

Table 2.2 Difference of the factors in a general recommender system and an educational recommender system

A significant number of recommender systems have been proposed in the education domain, as well as in teaching and academic guidance. In the education domain, the target users are students, teachers or academic advisors and the recommendable items are educational materials, universities or information such as courses, topics, student performance and the field of study.

In this thesis, to evaluate the proposed methodology, the focus was how to reduce information overloading for the student who is required to make a decision and thus to provide them with the most appropriate university course. The word “course” in the thesis refers to the programme of studies such as undergraduate courses or postgraduate courses. Many types of research have been undertaken, using different techniques and algorithms, that have been used to recommend courses to students. For instance, Sandvug and Burke presented the Academic Advisor Course Recommendation Engine (AACORN) that used a case-based reasoning approach which utilised knowledge acquired from previous cases in order to solve new problems (Sandvig & Burke, 2005). Their system used both the course histories and the experience of past students as the basis of assisting future students in course selection decision making.

At the same time, it was noticed that the intended future career of students was an essential factor which could influence their decision to choose a particular course (Huang, 2017). Farzan and Brusilovsky proved this by using a reported course recommendation system based on an adaptive community (Farzan & Brusilovsky, 2006). They employed a social navigation approach to analysing the students’ assessments of their career goals in order to provide

recommendations for courses. The main object of this approach was to obtain the students' explicit feedback implicitly, as part of their natural interaction with the system.

In this respect, Artificial Intelligence techniques could help to develop and improve the decision making and reasoning process of humans to minimise the amount of uncertainty there is in active learning to ensure a lifelong learning mechanism (Bobadilla, Ortega, Hernando, & Gutiérrez, 2013). The challenge for recommender systems, therefore, is to better understand the student's interest and the purpose of the domain (Ibrahim, Yang & Ndzi, 2017). An association mining based recommender system has been developed for recommending tasks that are related to learning which is most suitable for learners based on the performance of the targeted student and other students who are similar to them (Noakes et al., 1968). A course recommendation system has been proposed that would check how similar university course programmes are to the students' profiles.

The proposed framework in this thesis is comprehensive in that it combines content based filtering and collaborative filtering with an ontology technique in order to overcome the problem of overloading information. It does this by utilising a similar hierachal ontology to map the course profiles with the user (student) profile. The new approach develops two novel methods to extract and integrate data from multiple sources and then align the data appropriately. This ontology mapping of the different data improves the ability to obtain a comprehensive knowledge of the recommended items. The approach tackles the new user problem by calculating the ontology similarity that exists between the users' profiles by measuring the user rates for each item. The proposed recommender system is used to evaluate the hierarchy ontology similarity there is between the item profiles and the users' profiles before the student chooses a course to match his/her requirements and enrols on the programme.

### **2.2.1 Course Recommender System**

Research shows that students are overloaded by a large amount of information available when choosing a course (Huang et al., 2013). Nowadays, the range of course-related information available to students is still rapidly increasing (Diem, 2015). This abundance of information has created the need to help students to choose, organise and use resources that match their individual objectives, interests and present knowledge (Farzan & Brusilovsky, 2006). Dr Cable, a British politician who was the Secretary of State for Business, reported, "Applying to university is a big decision, and we want to ensure that all students, whatever their background, have the key facts at their fingertips to help them make the right choice for them". Bendakir

and Aïmeur report students pursuing education are faced with two challenges: a myriad of courses from which to choose, and a lack of knowledge about which courses to follow and in what sequence (Noakes et al., 1968). Also, the heterogeneity of course information and the user's personal needs make the decision process very tedious and complicated (Ge, Chen, Peng, & Li, 2012). At the same time, Wendy Hodgkiss, a careers adviser with the 'Which? University' organisation reported, "Don't just 'grab' something in a panic. Do some research first and make sure you really want to go to the relevant university and course"<sup>1</sup>.

Amer and Jamal showed that course choice decision is influenced by the background of the student and personal or career interests (Al-Badarenah & Alsakran, 2016). However, offering more course information on university websites does not necessarily infer that the students will have the cognitive ability to evaluate them all as alternatives. Instead, it confronts students with a problem usually termed as "information overloading" (Huang et al., 2017; Obeid et al., 2018; Shrivastav & Hiltz, 2013; Sieg et al., 2010). Evaluating all course alternatives themselves is a challenging task for students, even when some search tools do exist. How to automatically find the relevant course to match with the students' needs is a pressing problem (Ibrahim et al., 2018).

Many online systems have been made available to help students to find and compare different courses across different universities such as UCAS<sup>2</sup>, ukcoursefinder<sup>3</sup>, unistats<sup>4</sup> and comparetheuni<sup>5</sup>. However, these tools have been recognised as working based on previous user's knowledge of courses only or on keyword-based queries. Furthermore, for a given course query, a student will receive hundreds of results in a random order thus, again, students will be overloaded with information and potentially irrelevant results.

Good progress has been made in the course recommendation system which aims to support students to find suitable courses. Excellent work has been achieved in building a course recommendation system based on a collaborative filtering approach (Carballo, 2014; Ray & Sharma, 2011). In addition, CourseAgent is a significant work which is a community-based course recommendation system that uses the social navigation approach (Farzan & Brusilovsky, 2006) to produce recommendations for courses based on a student's estimation of their appropriate career goals. The main focus of this method is to collect explicit feedback from students implicitly, as part of their natural communication with the system. The basic and obvious advantage of the system to students is as a course administration system that stores information about courses they have chosen and facilitated communication with their advisors.

<sup>1</sup> retrieved from <http://www.hefce.ac.uk> at 25 Oct. 2018

<sup>2</sup><https://www.ucas.com>

<sup>3</sup><https://unistats.direct.gov.uk>

<sup>4</sup><http://comparetheuni.com>

<sup>5</sup><http://www.ukcoursefinder.com>

Sandvig and Burke reported a new course recommendation system called AACORN, which is a case-based reasoning approach(Sandvig & Burke, 2005). The system utilises the experience of previous students and their course histories as a place to start to advise course selection. In order to discover the similarities between course histories, the system uses a metric broadly used in bio-informatics named the edit distance. The system demands details of a partial history of the courses taken by a student before it can supply effective recommendations.

The RARE recommender system combines association rules with user preference data to recommend relevant courses (Noakes et al., 1968). RARE was used on real data derived from the Department of Computer Science at the Universite de Montreal. It analyses the past behaviour of students regarding their choice of course. More explicitly, it formalises association rules that beforehand were implicit. These rules allow the system to predict recommendations for new students. A solution to handle the cold start problem, which is central for recommender systems, is also proposed in RARE.

PEL-IRT refers to the Personalised E-Learning system which applies item response theory (Chen, Lee, & Chen, 2005). It recommends suitable course material to students, bearing in mind both the difficulty of the course material and student ability. When utilising PEL-IRT, students can choose course categories and units and can use relevant keywords to search interesting course material. Once the course material has been suggested to students, and they have browsed through the information, the system requires them to answer two questionnaires. This explicit feedback is employed by PEL-IRT to re-evaluate the students' abilities and to customise the course material difficulty featured in the recommendation.

Recently, academics have found that personalisation is an influential factor used to increase the accuracy of recommendations and information retrieval (Huang et al., 2017; Salahli, Özdemir, & Yaşar, 2013). Punj and Moore realised that recommendation agent that can filter and integrate information and offer feedback influenced the user's decision more in comparison to agents that are only aware of the alternative options (Punj & Moore, 2007). Furthermore, it was realised that the relevant course recommendations outcome integrates useful information from multiple useful sources such as jobs sites, social networks and other related educational data sources (Huang et al., 2013).

The Course Recommender System is based on several different collaborative filtering algorithms such as user-based (Cone, 2011) and item-based (Sarwar et al., 2001), OC1 (Murthy, Kasif, Salzberg, & Beigel, 1993). The Course Recommender System (Mahony & Smyth, 2007)

is based on a variation of the widely-used item-based collaborative filtering algorithm. The purpose of a module recommender system is to facilitate and enhance the online module selection process by recommending optional modules to students based on the core modules that they have selected. Applying historical enrolment data points leads to a very encouraging performance concerning both recall and coverage. Certain recent research has focused on using course recommender systems in niche areas such as civil engineering professional courses (Zhang, 2009) and physical education courses at universities (Liu, Wang, Liu, & Yang, 2010). From a study of the literature, it is obvious that the recommendation technology applied in the field of education can facilitate the teaching and learning processes. Considering the significance and importance of education, the assistance of a recommendation system can improve efficiency and increase the validity of learners in the actual educational situation.

All the above studies highlight the importance of course recommendation systems. However, all the current systems that provide information regarding a suitable course for students use a single data source such as the students themselves, courses histories or university records. However, the proposed search in this thesis seeks to build a new course recommendation system based on integrating information about courses from multiple data sources such as university websites, job websites and social networks. It will provide the students with recommendations that meet their personal needs, interests and career aspirations and therefore support the decision making process. At the same time, it will reduce information overload and heterogeneity through the use of ontology-based data integration of the search results of the relevant course. Accordingly, this creates the need for software that can automatically avoid the irrelevant choices, gathering information about the choices and allowing users to see only the more appropriate options that best match their needs.

Nevertheless, despite the high impact of course recommendation systems and how useful they are, there remain certain significant limitations, such as:

- Models based mainly on keywords fail to address the individual user's needs in the recommendation process. Although models use collaborative filtering and data mining such as association rule and decision trees, there is often a lack of historical information that makes it challenging to use this approach. For instance, new students who want to use the systems cannot generate any recommendations since the system has insufficient information about them.

- The shortcoming of models that use content-based filtering is that current approaches are based only on a specific subject recommendation rather than a whole university course. Moreover, the similarity calculation in these models is based on the weighted average of features and does not take into account user interaction with the system, such as the rating value of recommendation items.
- An additional shortcoming of the current models is that they do not provide comprehensive knowledge about the course that is most relevant to the student. For example, students need to know what future career the course will lead to and require information about this aspect, as well as the quality of the facilities of the educational institution that will be providing the course.

By categorising the needs of students and their areas of interest, it is possible to recommend an appropriate course. It is possible to help students to select a course by developing methods that will both integrate the data from multiple heterogeneous data sources and allow this to rapidly set valuable course-related information (Huang et al., 2013). By using ontology, the user will be able to obtain precise knowledge about the course (Wang & Sapporo, 2006). It is possible to build a relationship between the relevant information on the internet including the course modules, job opportunities and the users' interests. Ontology provides a vocabulary of classes and properties that can be used to both describe a domain and emphasise knowledge sharing (Hongji Yang, Zhan Cui, & Brien, 1999). The use of semantic descriptions of the courses and the student profiles allows there to be both qualitative and quantitative reasoning about the matching, as well as the required information about the courses and the student's interests, which is required in order to refine the selection process of which course to take.

A novel hybrid filtering system is proposed in this thesis, based on both the content based and collaborative filtering methods and using an ontology as the way to overcome the problem of information overloading which has been a key challenge when facing the building of an effective recommendation system. The proposed approach uses an ontology for data extraction and integration from multiple data sources. Data integration that is based on ontology is used in the ontology-based metadata. It uses a combination of model-based and memory-based ontology in collaborative filtering to provide a high-quality recommendation.

User profiling that is based on ontology, item ontology, the semantic similarity between two ontologies and the proposed OKNN algorithm is used in the collaborative filtering aspect to overcome the new user problem (more details of OKNN are provided in chapter 4). On the

other hand, item-based ontology and semantic similarity are both applied in content-based filtering to overcome the new item cold start problem. In order to make the measurement of semantic similarity more accurate, a hierarchy concept similarity approach is used in the content-based filtering. This measures the “IS-A” degree between the two nodes of item ontology which was found to yield a more precise recommendation list for the target user.

### **2.2.3 Challenges in the course recommendation system**

There are many challenges that it is necessary to overcome for the implementation of education recommender systems to be effective. In this section, the main challenges found in the literature and ways to address these are summarised.

**Information overload:** This problem refers to when, in an environment such as the internet, the amount of pedagogical content is overwhelming and widely spread over the generating network (Wang, 2008). This leads to information overload, making it difficult for students to find and evaluate quality information regarding the most suitable learning resources (Gasparini, Lichnow, Pimenta, & De Oliveira, 2009; Yang & Wu, 2009).

**Lack of structure in the data:** One of the primary difficulties and essential characteristics of a recommender system is the process by which the data is structured. (Pearce, 2008). For example, in social learning environments, information tends to be classified in just one category thereby reducing the number of options to the user (Abel et al., 2008). In addition, a predefined structure does not exist in a social network (Nikos Manouselis, Drachsler, Verbert, Santos, & Konstan, 2014) and information cannot be reused in other systems because of a lack of structure which hinders the interoperability among recommender systems (Nikos Manouselis & Vuorikari, 2009).

**New user and cold start problem:** This problem occurs when there are no ratings for new resources or when a new user has not yet rated any items (Tang & McCalla, 2009a). Suggested ways to overcome this challenge include 1) a knowledge provider can be the first starter; consequent users can contribute to this elaboration (Rafaeli, Dan-Gur, & Barak, 2005); 2) use of artificial learners (Tang & McCalla, 2009a); 3) use of information related to the completion of activities and similar preferences (Manouselis et al., 2014).

**Cognitive overload:** This refers to the effort required during the process of selecting useful resources or assigning accurate ratings (Rafaeli et al., 2005). This takes place particularly when there is raw data, when the user is unable to ask the right question, when pedagogical resources

are not properly defined by the expert (DeLong, Desikan, & Srivastava, 2006), when resources are not classified and when there are no existing summaries, keywords or other types of descriptors (Yang, Huang, Tsai, Chung, & Wu, 2009). One way to overcome this issue is to use content analysis techniques such as data mining to find keywords or structures (Yang et al., 2009).

**Quality of the recommendation and trust:** Another problem arises when users do not trust the system and the recommendations. The probability that a user will perform an action based on the recommendations is often too low (Chang et al., 2005; Schulz et al., 2001). For that reason, it is suggested that the quality of a recommendation is always defined (Zheng & Li, 2008), that it should be made clear whether a recommendation is either precise or simply relevant, that biased recommendations are reduced as much as possible (Schulz et al., 2001) and that it is made clear where the recommendations come from (Rafaeli et al., 2005) or how new items are added.

## 2.3. Ontology

Modelling information at the semantic level is one of the main purposes of using ontologies (Guarino et al., 2009). This section gives a detailed review of ontology including ontology definitions and a description of the ontology development process. It then discusses the application of ontology to data extraction and integration and the use of ontology in recommender systems.

### 2.3.1 Ontology Definition

The original definition of “ontology” in computer science was provided by Gruber (Gruber, 1993) as an “explicit specification of a conceptualisation”. Borst defines ontology as a “formal specification of a shared conceptualisation” (Borst, 1997). Coelho *et al.* gave a new definition of ontology “as a knowledge domain conceptualisation into a computer-processable format which models entities, attributes, and axioms”. Ontology is typically made up of vocabulary and relationships between concepts (Coelho, Martins, & Almeida, 2010). According to Antoniou and Harmelen (Antoniou & Harmelen, 2008), ontologies are concept properties, disjointedness statements, value restrictions and specifications of logical relationships between objects. Ontologies have provided a tool for formally modelling the structure of a system based on the relationships that emerge from its observation.

The term taxonomy is used when the ontology contains only “IS-A” relationships, and normally the use of the word ontology is restricted to systems that support a rich variety of relationships between the concepts, including logical propositions that formally describe the relationship. Many ontology classifications have been established (Grazia, Bono, Pieri, & Salvetti, 2004). For example, ontology can refer to the specific domains that can provide conceptual modelling of a particular domain.

Ontologies can be classified into three categories - Domain Ontology, Upper Ontology and Application Ontology. Domain Ontology represents the vocabulary related to a generic domain such as education, medicine or automobiles; or any generic task or activity such as selecting or diagnosing by specialising the terms introduced in the top-level Ontology. Upper Ontology, also called Foundation Ontology, is a model of the common objects that can apply to a wide range of Application Ontology. An Application Ontology defines concepts of a particular domain and task. In the application domain, Upper Ontology, as well as Domain Ontology, can be integrated with Application Ontology. Practical descriptions on ontologies have shown their importance in several respects:

- An ontology involves the factorisation of knowledge. Like the oriented object approach, knowledge is not repeated in each instance of a concept (Rinku & Aravind, 2016).
- An ontology provides a unified framework to reduce or eliminate ambiguities and conceptual and terminological confusion (Daramola, Adigun, & Ayo, 2009).
- An ontology can significantly increase the performance of search engines. Through the semantics provided, an ontology can address problems such as the noise and silence of the traditional search engines (Ringe & Francis, 2012).
- An ontology can support the sharing and reuse of knowledge (Shvaiko & Euzenat, 2013). the researcher can reuse existing ontologies and, if adapted to meet with their need, will reduce the time of creating an ontology from scratch.
- An ontology implements mechanisms of deductive reasoning, automatic classification, information retrieval and ensures interoperability between systems.

In the following sections, ~~this discussion~~ will focus on all of these aspects of the ontology domain. The importance of ontology in the field of education will be explained, and the architecture of the framework and the procedure of acquiring and representing an ontology in that domain will be specified. Ontologies are classified according to their level of dependence

on a particular task or point of view into three categories (Kawtrakul, Suktarachan, & Imsombut, 2003):

- Upper-level ontologies are domain independent and intended to capture and represent the semantics of the real world to support large applications. An example of this type is the “Cyc project” which attempts to capture and encode large amounts of common sense knowledge about the real world.
- Domain ontologies specify concept relationships between concepts and inference rules for specific domains in a specific way (e.g. travel reservations, soccer and gourmet food).
- Application ontologies describe concepts relative to a task domain such as the reasoning process for medical diagnosis. According to the classification mentioned above, the ontology developed in this thesis is categorised as an application ontology which is to be utilised within the e-Government service domain.

In this thesis, ontology has been used in the three main areas of the proposed framework - the data gathering component used an ontology to extract information from multiple sources based on the hierarchy structure of information, the ontology model component, used to construct and map ontologies, and the recommendation engine component. The following subsection describes the background to using ontology in the relevant domain of this thesis.

### 2.3.1 Ontology Representation

The main principal elements in ontologies are concepts, relations, axioms and instances. The definition of each element, according to (Noy et al., 2001), is presented below:

A **Concept** (also known as a term or a class) is the essential abstract component of a domain. Typically, the class represents a group of common properties owned by many members. Also, classes are arranged in hierarchical graphs on two levels. Higher level classes are called parent classes, and the subordinate levels are called child classes. A graph of concepts might organise classes in a lattice or a taxonomic view; for example, the class “Faculty” could have many subclasses, such as “Department” and “College”. Moreover, the concepts might have many different distinguishable properties.

A **Relation** (also known as a slot) is used in the ontology structure to provide a declaration for the relationships between concepts in a specific domain. In order to specify the two classes involved in a particular relationship, one of them will be described as a “Domain” and the other

one as a “Range”; for instance, the relationship “Work” can have the concept of “Employee” as a domain and “Faculty” as a range.

An **Axiom** (sometimes called a facet or role restriction) is utilised in the ontology to force restrictions on the values of both classes and instances. Logic-based languages, such as first-order logic, have been developed in order to express these constraints. Furthermore, these languages can be used as the verification process for the consistency of the ontology structure.

An **Instance** (also known as an individual) is a relationship between ontology concepts in relation to their real values; for instance, “Iraq” could be an instance of the class “Asian countries”, or simply “countries”.

### 2.3.2 Data Extraction Using Ontology

Vast amounts of information can be found on the web (Vallet *et al.*, 2007). Consequently, finding relevant information may not be an easy task. Therefore, an efficient and effective approach which seeks to organise and retrieve relevant information is crucial (Yang, 2010). With the rapid increase of documents available from the complex WWW, more knowledge regarding users’ needs is encompassed. However, an enormous amount of information makes pinpointing relevant information a tedious task. For instance, the standard tools for web search engines have low precision as, typically, some relevant web pages are returned but are combined with a large number of irrelevant pages mainly due to topic-specific features which may occur in different contexts. Therefore, an appropriate framework which can organise the overwhelming number of documents on the internet is needed (Pant *et al.*, 2004).

The educational domain is one of the domains that have been affected by this issue (Almohammadi, Hagras, Alghazzawi, & Aldabbagh, 2017). As the contents of the web grow, it will become increasingly challenging, especially for students seeking to find and organise the collection of relevant and useful educational content such as university information, subject information and career information(Chang *et al.*, 2016). Until now, there has been no centralised method of discovering, aggregating and utilising educational content (Group, 2009) by utilising a crawler used by a search engine to retrieve information from a massive number of web pages. Moreover, this can also be useful as a way to find a variety of information on the internet (Agre & Dongre, 2015). Since the aim is to find precise data on the web, this comprehensive method may not instantly retrieve the required information given the current size of the web.

Most existing approaches towards retrieval techniques depend on keywords. There is no doubt that the keywords or index terms fail to adequately capture the contents, returning many irrelevant results causing poor retrieval performance (Agre & Mahajan, 2015). In this thesis, a new approach to web crawler based on ontology is proposed which is used to collect specific information within the education domain. In this thesis, the approach focuses on a crawler which can retrieve information by computing the similarity between the user's query terms and the concepts in the reference ontology for a specific domain. For example, if a user seeks to retrieve all the information about master's courses in computer science, the crawler will be able to collect all the course information related to the specific ontology designed for the computer science domain.

The crawling system described in chapter 3 matches the ontology concepts thus giving the desired result. After crawling concept terms, a similarity ranking system ranks the crawled information. This reveals highly relevant pages that may have been overlooked by focused standard web crawlers crawling for educational content while at the same time filtering redundant pages thereby avoiding additional paths.

### 2.3.3 Ontology Mapping

Ontology mapping is also known as ontology matching or ontology alignment. Ontology mapping or matching is different from ontology merging. Ontology mapping tries to make the source ontologies consistent and coherent with one another while keeping them separate. In contrast, ontology merging aims to create a single coherent ontology that includes the information from all the sources. Ontology mapping is used to “establish correspondences among the source ontologies, and to determine the set of overlapping concepts, concepts that are similar in meaning but have different names or structure, and concepts that are unique to each of the sources” (Noy et al., 2001). It is also defined by Kalfoglou and Schorlemmer (Kalfoglou & Schorlemmer, 2003) as follows, “Given two ontologies O1 and O2, mapping one ontology onto another means that for each entity (concept C, relation R, or instance I) in ontology O1, it tries to find a corresponding entity, which has the same intended meaning, in ontology O2”. This research defines ontology mapping so as to find a set of semantic correspondences between similar elements in different ontologies.

Various works have been developed to support the mapping of ontologies. An interesting survey which gathered more than 30 works is presented in (Kalfoglou & Schorlemmer, 2003). In(De Bruijn et al., 2006) other surveys can be found regarding ontology alignment. In most

approaches, heuristics are described for identifying corresponding concepts in different ontologies, e.g. comparing the names or the natural language definition of two concepts and checking the closeness of two concepts in the concept hierarchy. PROMPT (Natalya & Musen, 2004) is an algorithm for ontology merging and alignment based on the identification of matching class names. A few approaches, such as RDFT (Omelayenko, 2002), use the comparison of the resources to determine a similarity between concepts, but the problem is that the structures of all data instances are heterogenous. RDFT proposes an approach to the integration of product information over the web by exploiting the data model of RDF which is based on directed label graphs. RDFT discovers a similarity between classes (concepts) based on the instance information for this class, using a machine-learning approach. Like RDFT, GLUE (Kalfoglou & Schorlemmer, 2003) is a system which employs machine learning technologies to semi-automatically create mappings between heterogeneous ontologies. An ontology is considered here as a taxonomy of concepts, and the problem of matching is reduced to “for each concept node in one taxonomy, find the most similar node in the other taxonomy”. The problem with GLUE is that the reliability of the results is related to the quantities and the degree of correction of all examples used by machine learning. S-Match Semantic Matching (Giunchiglia, Shvaiko, & Yatskevich, 2004) is an approach to matching classification hierarchies. Semantic matching addresses the problem of when there are two different classification hierarchies, where each hierarchy is used to describe a set of documents, i.e. each term in the classification hierarchy describes a set of documents.

## 2.4 Summary

This chapter described related works and concepts that have been discussed in this thesis. Recommended systems (RSs) provide a promising approach to information filtering as they help users to find the most appropriate items. Based on the needs of each user recommendation system, a series of specific suggestions will be generated. It's highlighted the main recommendation approaches and explained the principles of similarity calculation of each approach. In addition, the drawback and advantage of each approach has been detailed.

Despite the high impact of the course recommender system and how useful it is, there are certain significant limitations in the current researches and approaches. Approaches based mainly on the keywords failed to address the individual user's needs in the recommendation process. Although models use collaborative filtering, and data mining such as association rule and decision tree, there is often a lack of historical information that makes it challenging to

adopt this approach. For instance, new students who wish to use the systems do not have sufficient information about the model and therefore cannot generate any recommendations. The shortcoming of approaches that use content-based filtering is that current approaches are based only on a specific subject recommendation rather than an entire university course. Moreover, the similarity calculation in these models is based on the weighted average of features and does not take into account user interaction with the system, such as the rating value of recommendation items. Another shortcoming of the current models is that they do not provide comprehensive knowledge about the course that is most relevant to the student. For example, students need to know what future career the course will lead to and require information about this aspect, as well as the quality of the facilities of the educational institution itself that will be providing the course.

Our focus in this thesis were how to apply content based approach and collaborative based approach utilising ontology in education domain. A novel hybrid filtering approach is proposed in this research, based on both the CBF and CF methods and using ontology as a way by which to overcome the problem of information overloading which has been a key challenge when consideration is given to building an effective recommendation system. The research used ontology for data extraction and integration from multiple data sources. Data integration that is based on ontology is used in the ontology-based metadata. It utilises a combination of model-based and memory-based use of ontology in CF to provide a high-quality recommendation. User profiling based on ontology used in the CF to overcome the new user problem. On the other hand, item-based ontology and semantic similarity are both applied in CBF to overcome the new item cold start problem.

# **CHAPTER 3 OPCR: ONTOLOGY-BASED PERSONALISED COURSE RECOMMENDATION FRAMEWORK**

*"Customers don't know what they want until we've shown them"*

Steve Jobs

## **3.1 Introduction**

Choosing a higher education course at university is not an easy task for students. A wide range of courses is offered by individual universities whose delivery mode and entry requirements all differ. Finding relevant information regarding higher education from a large number of websites is a challenging and time-consuming process. Helping students to make the correct choice from a myriad of available courses in order to meet their individual needs is a testing experience. Such abundant information means that students need to search, organise and use the resources that can enable them to match their individual goals, interests and current level of knowledge appropriately. This can be a time-consuming process since it involves accessing each platform, searching for available courses, carefully reading every course syllabus and then choosing the one that is most suitable for the student. However, simply because more course information is now provided by universities on their websites does not automatically mean that students possess the cognitive ability to evaluate each of the courses. Instead, they are confronted with a problem that is termed "*information overloading*". To counter this, artificial intelligence methods are now being applied to information retrieval systems. Recommendation systems provide a promising approach to information filtering as they help users to locate the most apposite items. Based on the requirements of each user's recommendation system, a series of specific suggestions can be generated. Thus, a personalised recommendation system can be an effective way of suggesting relevant courses to prospective students.

There are many online systems currently available that can be used to find and search for courses which use tools based on the users' prior knowledge of the courses and keyword-based queries. However, these approaches fail to address the needs of an individual user in the recommendation process. Moreover, the models use collaborative filtering and data mining, such as association rule and, as there is often a lack of historical information, this makes it challenging to adopt these approaches. For instance, new students who wish to use the systems do not have sufficient information about the model, and therefore no recommendations can be provided. On the other hand, the approaches that use content based filtering focus only on a

specific subject recommendation rather than an entire university course. Moreover, the similarity calculation in these models is based on the weighted average of features and does not take into account a user's interaction with the system, such as the rating value of the recommended items.

This chapter presents a novel approach that personalises course recommendations so that the individual needs of users are suitable matched. The proposed approach has developed a framework of an ontology-based hybrid filtering system, the ontology-based personalised course recommendation (OPCR). OPCR is a modular framework for the creation of knowledge-based recommender systems that utilise ontology as their source of knowledge. The motivation behind this approach is to overcome not only the limitations that experts impose on the automation and maintenance of such systems but also to address the cold start problem experienced by a new user of the system by using ontology matching between both the user profile and the course profiles. The proposed architecture makes use of AI to obtain the required information, where possible, in an effort to minimise the tasks required by the ontology. OPCR tackles the problem of information overloading by actually limiting the available courses the student has to examine as possible choices. In addition, OPCR uses dynamic ontology mapping between user profiles and courses profiles that lead to a reduction in the time taken to search relevant courses and improves the performance of the system.

A hybrid recommender method based on ontology is proposed in this work. The approach firstly aims to extract and integrate information from multiple sources based on ontology. The information sources are then classified into three primary sources; course information sources, student information sources and career information sources. Integrating this information using ontology will obtain optimal results.

Moreover, the second objective is to build dynamic ontology mapping between the user profiles and the item profiles that will help to reduce information overloading. In order to offer an appropriate recommendation to users, two main filtering approaches, CBF and CF, have been combined and the result is thus a combination of memory-based and model-based methods. In CF, several techniques, such as user profiling that is based on ontology, item ontology and KNN are used to overcome the information overload problem and to improve scalability and accuracy.

On the other hand, item-based ontology and semantic similarity are applied in content-based filtering to solve the new user issue and also to improve accuracy. The final objective is to put

forward a list of recommendations and ask the user to assign a rating to each recommendation. The user then provides feedback on the recommendation list and carries out a re-ranking. User feedback has been used to evaluate the system and to improve its accuracy, as is shown in greater detail in the evaluation chapter. This work aims to increase the accuracy and performance of the recommender system by combining the hybrid method (CBF and CF) with enhanced ontology.

## 3.2 Framework Architecture Design

OPCR has been built for situations where there is a need to identify a relevant university course program for a particular student. Within its scope, the definition of what is relevant, as well as the possible use of the qualification, is defined by how the course will meet the individual needs of a student. The framework is extremely flexible since it can be adapted to any item domain that meets the specific requirement of having 'objectively' relevant items. Although the difference might not be immediately obvious, often the 'relevant' course is selected on a completely subjective basis, such as when selecting which movie to watch at the weekend or when buying books or clothes. Making a generalised suggestion in these circumstances can be unfounded but the situation is very different when choosing a university course programme such as BSc, MSc and so forth. During course selection, one can find specific features to quantify, for example, the ranking or location of the university. In addition, the intended application of a degree defines the entry requirements for courses, the course fee should match the budget of the student, and the course units/modules should be relevant. The other important factors with regard to the selection of a course are the location of the university providing the course and the type of employment that will be available following completion of the course. All these factors will impact on the decision making process when choosing the most appropriate university course. The factors will be different for each student based on their personal requirements. OPCR is therefore designed with these pertinent factors in mind.

The proposed ontology-based personalised course recommendation framework (OPCR) focuses on recommending courses to students by utilising a hybrid filtering approach that combines both content-based filtering and collaborative based filtering with ontology support. As shown in Fig.3.1, OPCR consists of four main layers. The first layer, data gathering, consists of all the information resources and the data collection module. This is used to extract useful information from multiple sources. The second layer is the database that is used to store all of the items and user information. The middle layer is the core functional part that includes the

ontological data model and the recommender engine. Each of these components will explain in detail in the following sections. The final layer is the user application layer, consisting of the user interface, which is responsible for user interaction with the framework, for searching items and for providing feedback on the recommendation list. Every layer and module in the framework both links and interacts with the others, based on the input and output of each one. The framework comprises the following steps:

- (1) Extract all the useful information from multiple sources for the system.
- (2) Build profiles of the courses by extracting all the useful information regarding course features and organising that information in the system database. Consideration is given to the ontology hierarchy of the course features.
- (3) Build the student profile by obtaining student information via both explicit and implicit approaches. Different user attributes have been identified which can be used to profile the student into the OPCR system as well as the user ratings of the recommended courses.
- (4) Build dynamic ontology mapping in order to link the user profile and the item profile.
- (5) Analyse user queries and calculate the similarity between the user profile and the course profile by employing ontology matching and cosine similarity.
- (6) Use a collaborative filtering technique in order to obtain top N users that are similar to the current user by using an ontology-based k nearest neighbour (OKNN) algorithm.

The final step suggests the recommended list of courses to the user and obtains feedback from the user. The purpose of each of these steps is explained in the following sections. All the modules, which have been fully developed using Open Source Free Technologies (OSFT), are organised in a traditional client-server structure. The most novel aspect of the system is the careful combination of different technologies, which has led to the development of an application that uses advanced artificial intelligence techniques in an efficient way, presenting a low execution time. These techniques are totally hidden from the users who simply interact with a user-friendly client application that presents information on maps and lists that are very easy to manage.

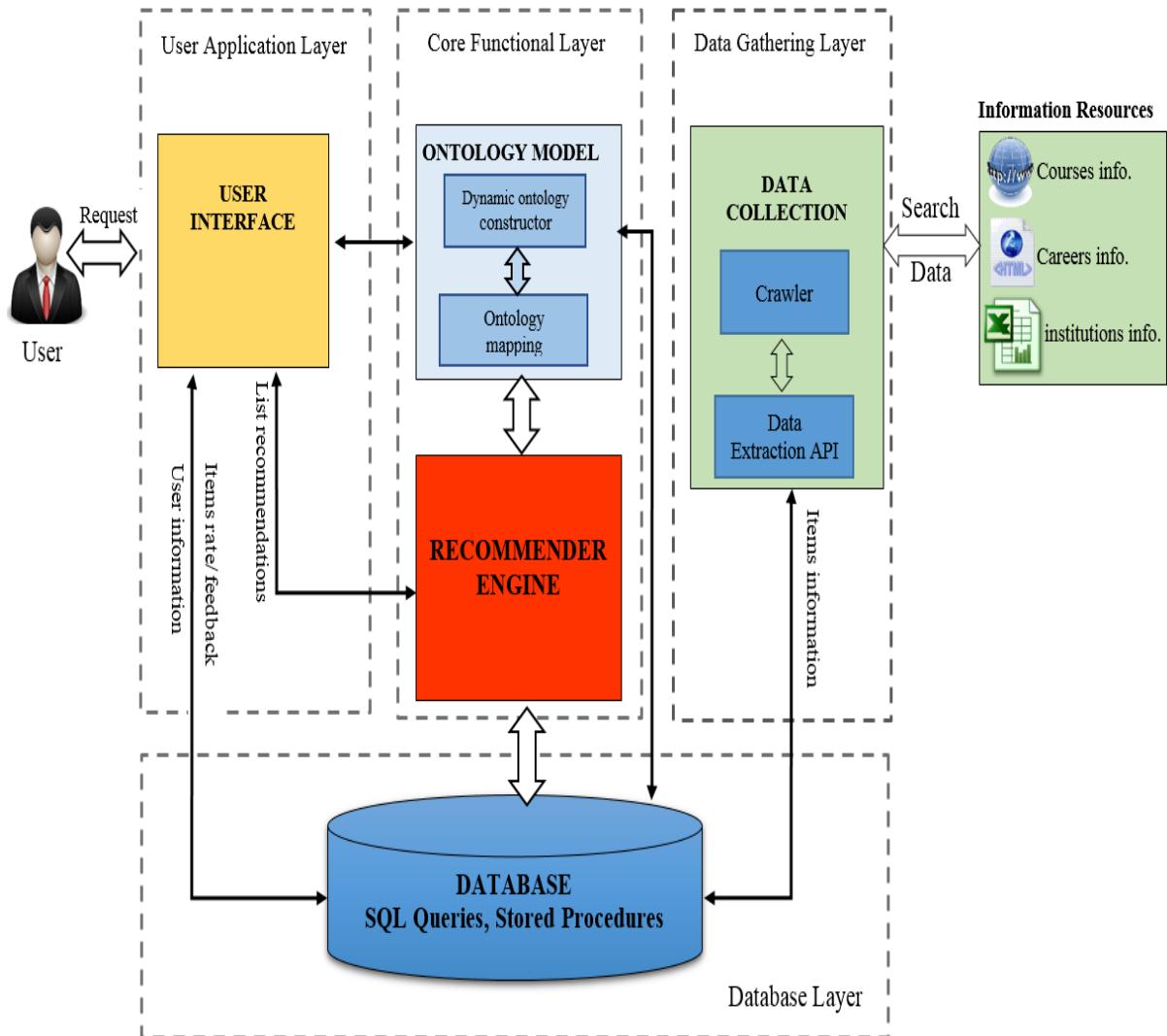


Figure 3.1 OPCR main architecture

The modularity of the framework allows components to be swapped with virtually no modifications required to other parts of the system. For instance, the web crawler functionality could be swapped with a pre-existing database, provided that the database contained all the required information. This also enables the independent modification and extension of each component. For example, the Data Collection is responsible for extracting the course data from a webpage but is so flexible that it can be used to extract any data in a different domain with only certain adaptations to the item attributes. Moreover, Ontology Model can easily be adapted for use in any domain. The following sections discuss each module in more detail.

### 3.3 Main Components

This section presented all the modules that have been developed for the framework from the server aspect that includes the data gathering module, the ontology model and the recommender engine as highlighted in Fig. 3.1. Each of these modules works sequentially and in correlation.

In the following subsections, the structure of each module is explained, and the input/output data for each is described.

### 3.3.1 Data Gathering

As it was decided that a content-based recommender system technique should be the primary approach for the provision of recommendations, different formats of information were required to be gathered to support this system. Fortunately, all of this information is available through sources that are publicly available. This includes websites in HTML format, such as the universities' websites for course information and recruitment websites for career information and Microsoft Excel documents that have been uploaded to the internet, such as statistical information regarding the reputation of educational institutions, for example, the NSS scores for universities. The data from both the student and course ontology is prepared and pre-processed into the correct format for the recommendation engine by the pre-processing data component. To obtain information about each course from the websites of all the universities was a time-consuming task as each university publishes its course information in a different format. Extracting precise information from various websites is always a challenging task in the domain of information engineering, so a web crawler was therefore customised that browses the web page automatically. The web crawler scrapes information from a web page and then sorts this into the system database. The reformulated queries are allocated to web crawlers and APIs that search for specific course information and jobs.

The web crawler analyses the web page based on a definition of the features of each course and then extracts the feature values. Each extracted feature value belongs to one of the features defined in this paper. Five features of the courses are used in this study: course title, course major subject, course fee, university location and the language of the course. On the other hand, the feature constructed in the user ontology is based on item ontology. The implicit information such as user feedback and the rates of the recommendations have been collected and added to the user profile for later use when it is then utilised to locate a top-rated neighbour that is similar to the target user.

The main challenge of the data gathering process was the building and customising of the web crawler to extract data from the web pages. All course and university information are available from the universities' websites. However, there is no suitable dataset available from which to generate a synthetic dataset to implement the proposed framework. Visiting the website of each

individual university and extracting information is a huge challenge and extremely time-consuming.

Moreover, the web pages of each university use a different layout, and the data is sorted in a particular format. This problem led to the use of the UCAS website in order to extract information regarding courses for universities throughout the United Kingdom. UCAS is one of the most popular higher education websites that details course information (undergraduate, postgraduate, etc.) for all UK universities. The challenge with using UCAS is that no API is provided with which to extract course information. This led to the building and customising of a data extraction API that could extract useful course information and save this in the system database to be used when needed for implementation and evaluation purposes.

Based on a similar concept, to extract job information the crawler was adapted and customised to obtain all useful information regarding jobs and save this in the system database. The Java technique was used to build the crawlers, and HTML was used to create the interface for the crawler. A new approach has been used in the data extraction module that extracts the data based on a hierarchy relationship between course information or job information. The idea is that, before extracting the information, the hierarchy relation between them needs to be defined as an example, the MSc course for computer programming will be defined as a subclass, with computer science defined as the main class. Furthermore, computer science will form part of information technology as a field of study. Extracting information based on a hierarchy relationship will help the system to avoid over-looping when creating a query in the database. The following subsections present further discussion regarding the data collection of course information, job information and other relevant information that has been used to improve recommendation quality. In Fig 3.2 main architecture of proposed crawler, the crawler consists of several stages; it begins with construction domain ontology which it uses as a reference of similarity between the user query and the web contents. The user query adjusts to generate query based ontology concepts and uses Term Frequency-Inverse Document Frequency (TF-IDF) for identifying terms for query expansion.

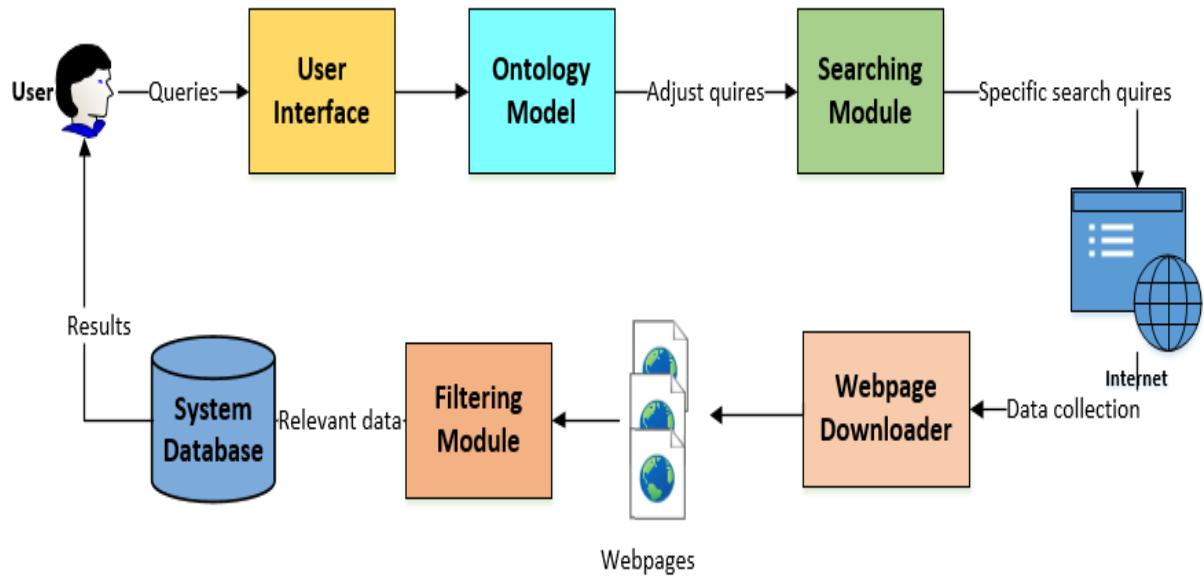


Figure 3.2 Proposed crawler main architecture

Firstly, we describe how information retrieval can be achieved in the ontology. For instance, if  $D$  is the number of documents annotated by concepts from an ontology  $O$ , the document is represented by vector  $d$  of concept weights. For each concept  $x \in O$  annotating  $d$ ,  $d_x$  is the importance of  $x$  in document  $d$ . It can be computed by using the TF-IDF algorithm as shown in Eq. 3.1:

$$d_x = \frac{freq_{x,d}}{\max_y freq_{y,d}} \log \frac{|D|}{n_x} \quad (3.1)$$

Where  $freq_{x,d}$  is the number of occurrences of  $x$  in  $d$ ,  $\max_y freq_{y,d}$  is the frequency of the most repeated instance in  $d$ , and  $n_x$  is the number of documents annotated by  $x$ , then cosine similarity between the query and the document is used as the relevance score for ranking the documents as shown in Eq.3.2.

$$\text{cosine similarity}(d, q) = \frac{\sum d_i \cdot q_i}{\sqrt{\sum d_i^2} \cdot \sqrt{\sum q_i^2}} \quad (3.2)$$

Where  $d$  the ith term in the vector for document and  $q$  the ith term in the vector for the query. The ontology-based query used as an input to the search engine module. The output of this phase is a set of documents which would be used for the crawling system and furthermore

operate as a way by which to check all the web pages for validity (i.e. HTML, JSP etc.). If it is valid, it is parsed, and the parsed content is matched with the ontology and, if the page matches, it will be indexed otherwise, it will be discarded. Architecture of the proposed approach is illustrated in Fig.1. The user interacts with the crawler using a simple interface designed to allow the user query insert.

### 3.3.1.1 Course Crawler Module

The web crawler has two main tasks, to acquire the web pages featuring relevant products and to extract from these the useful information that is needed for the recommendation process. The crawler was used to extract course information from the UCAS webpage. Both of the tasks of the crawler utilise the Document Object Model (DOM) that is used to describe all the elements of a webpage. As mentioned in the previous section, the primary source for university course information in the United Kingdom is UCAS. UCAS details more than 80,000 courses for different fields of study at different universities. In order to validate the framework, MSc courses will be the target of the crawler as a case study. Most students face difficulty in finding which master's programme is more relevant to their background or which will match the type of career they are considering. Through the webpage, the crawler can target specific objects in a webpage that might represent links or other relevant information. For example, the bullet point list of a course description is represented in the DOM by a list object.

Fig.3.3 shows a sample of a course page from the UCAS website. A sub-module of the crawler, the content extractor, adopts the same approach to identify the required information from the product webpage. The information required in this case is the unique DOM identification characteristic of the area that contains the required text. Four areas are extracted from each course webpage; the university name and course topic, the course details, the entry requirements and fees and funding.

This approach makes both the crawler and the content extractor very flexible since they can be customised for virtually any type of university course programme following a structural pattern. The content extractor can also be expanded to identify any other important areas of information that might be required in the future such as course modules/unit, assessment method and so forth. Furthermore, both components are virtually infallible. However, these advantages come at the cost of scalability since, for each new course that needs to be scanned, a new set of DOM characteristics needs to be identified. The flexibility that is required is also the reason why generic extraction frameworks have not been used. The outcome of the web crawler presents

courses names, accompanied by a detailed table of attributes that includes names, values and headings, as well as any possible bullet point descriptions. Table 3.1 shows the outcome of the web crawler for the course. Similar data tables are created for all scanned courses

The screenshot shows a course details page for 'Computer Network Administration and Management (Taught)' at the University of Portsmouth. The page includes sections for provider information, course contact details, open days, course details, and modules. The developer tools' Elements tab is open, displaying the underlying HTML code for the page, which includes various script tags, CSS classes, and server-generated data attributes.

Figure 3.3 HTML structure of a course webpage

Feature	Value
<b>Course title</b>	Artificial Intelligence
<b>Course qualification</b>	MSc
<b>Course URL</b>	<a href="https://digital.ucas.com/courses/details?coursePrimaryId=7fea172a-efdd-4c02-9950-edf34da09124&amp;academicYearId=2018">https://digital.ucas.com/courses/details?coursePrimaryId=7fea172a-efdd-4c02-9950-edf34da09124&amp;academicYearId=2018</a>
<b>Course description</b>	Artificial Intelligence (AI) forms part of many digital systems. AI is no longer seen as a special feature within software, but as an important development expected in modern systems. From word-processing applications to gaming, and from robots to the Internet of Things, AI tends to be responsible for controlling the underlying behaviour of systems. Such trends are forecast to grow further.
<b>University name</b>	University of Aberdeen
<b>Field of study</b>	Information technology
<b>Main subject</b>	Computer science
<b>Major subject</b>	Artificial Intelligence
<b>Course UK Fee</b>	£6,300
<b>Course international fee</b>	£15,000
<b>Course location</b>	University of Aberdeen

	King's Aberdeen AB24 3FX	College
<b>Entry requirements</b>	Our minimum entry requirement for this programme is a 2:2 (lower second class) UK Honours level (or an Honours degree from a non-UK institution which is judged by the University to be of equivalent worth) in the area of Computing Science. Key subjects you must have covered: Java, C, C++, Algorithms problem solving and Data Structures.	
<b>Course duration</b>	12 months	
<b>Course language</b>	English	
<b>Course mode</b>	Full-time	

Table 3.1 Example result of the web crawler for course information

### 3.3.1.2 Job Crawler Module

The second web crawler was used to extract job information from recruitment webpages such as Indeed.com. The job information crawler used the same principles as the course information crawler. For each job item that has to be scanned, a new set of DOM characteristics also needs to be identified. Fig. 3.4 shows a sample of a job webpage from the Indeed website. The crawler has been customised to obtain useful information about jobs featured by Indeed.co.uk, one of the most popular recruitment sites used to search for jobs in the United Kingdom. All information that the crawler extracted about the job is based on course majors. The information regarded as useful about the jobs was a job title, job description, company name, job location, job salary and reviews of the job. Table 3.2 shows the outcome of the web crawler for the job.

Feature	Value
Job title	Machine Learning/Artificial Intelligence Engineer
Company's name	EF
Job description	Machine Learning/Artificial Intelligence Engineer. We are looking for Machine Learning/Artificial Intelligence engineers to help us build the most intelligent system.
Job location	London
Job salary	£35,000 - £45,000 a year
Job education requirement	MSc Artificial intelligence
Job review	30 reviews

Table 3.2 Example result of the Web Crawler for job information

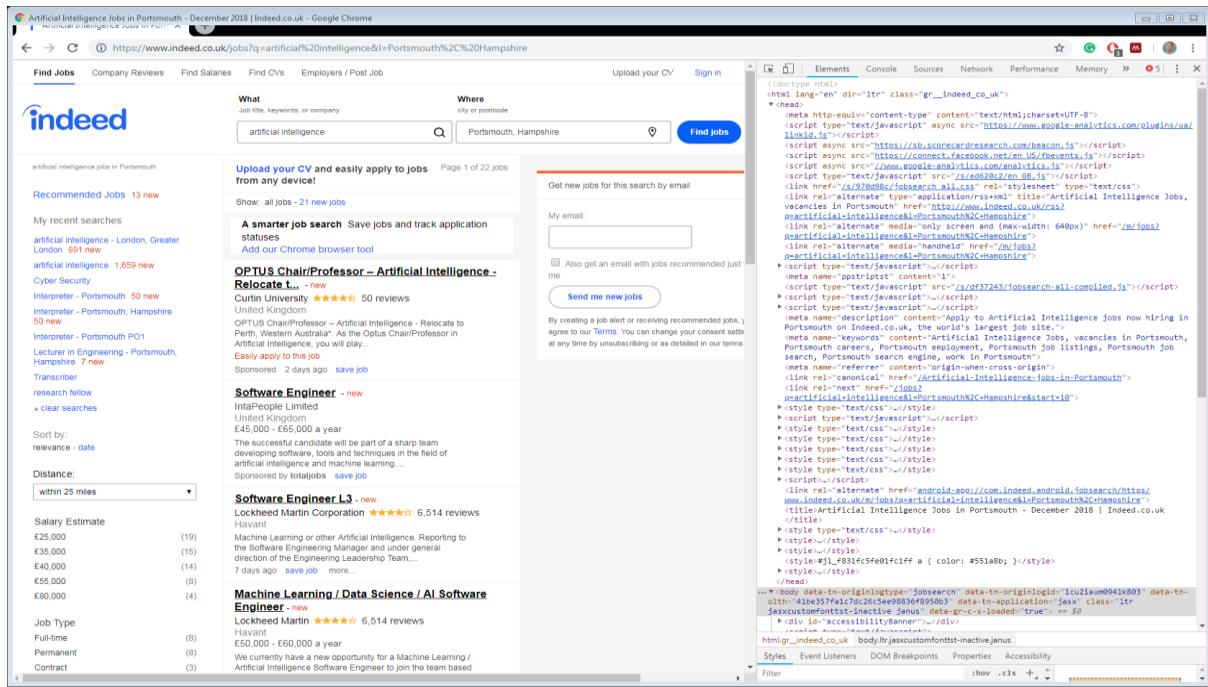


Figure 3.4 HTML structure of a job webpage.

### 3.3.1.3 NSS Score Data Collector

The national student survey (NSS) is one of the significant factors students have to consider when choosing a university course programme in the United Kingdom. NSS contains much statistical information regarding a university such as teaching average, student satisfaction, lab facilities and so forth. The NSS report is published every year with details of all educational institutions in the UK. The NSS score collector has extracted the NSS information and sorted this into SQL format in the system database.

### 3.3.1.4 University Rank Collector

One of the critical factors that can affect decisions when choosing a university course is the reputation of the university (Brown, Varley, & Pal, 2009). OPCR has extracted the rank for each university from the popular education website such as [theguardian.com](http://theguardian.com)<sup>1</sup> and sorted this in the system database in SQL file format. OPCR used the university rank as one of the factors in the final scoring function for recommendations to compute the similarity between each course in the database.

### 3.3.1.5 Feature Extraction

Identifying different attributes is necessary for course profiling (Lee, 2011). In order to construct a course ontology, the factors that most influence a student when making a decision

to choose a university course needed to be identified. These factors then formed the main classes of the ontology. A survey of students at the University of Portsmouth was carried out to discover the most important factors that had influenced their choice of university course. The programme title, fee, location and prominence of a course were all factors that appeared to be the most important when the students determined their choice of university for higher education (HE) study.

It was found that issues of institutional prominence maintain a fairly high profile in students' decision-making. The overall reputation of the institution and the National Student Survey score (NSS) of final year undergraduate students are both significant.

### **3.3.1.6 User Profile data collector (UPDC)**

This component is responsible for collecting user profile information either explicitly or implicitly. User information includes demographic data, academic information and career interests as well as information regarding the user rating of recommendation items. This information was extracted based on two approaches. The first approach was to ask the student information when he/she registered with OPCR as shown in Fig 3.1 OPCR has a user interface, which is web-based using HTML, that allows the user to make interactive by completing the user profile and providing feedback for the recommendations. The registration information for the user includes personal information such as name, age, gender and postal address together with academic information such as educational background, the field of study, interest area, skills and so forth. Information obtained by asking the user for this directly is referred to as explicit. The second approach, implicit information, is based on feedback received from the user regarding the recommended item and how they rated the recommendation.

## **3.3.2 Ontology Model**

With the growing need for more effective use of ontology in a wide range of applications, there has become an increasingly high demand for the creation of a suitable construction approach for ontology. Ontologies are constructed using two methods, i) the manual process (semi-automated) with a supervised approach; and ii) the automated process with an unsupervised approach (Zeng, Zhu, & Ding, 2009). Thus far, construction approaches for most of the applications have applied a semi-automated process. Ontologies are constructed manually using various methods such as machine learning techniques and data mining techniques that can build an ontology of high quality. However, from the perspective of performance, this can

be costly and time-consuming and to maintain and manage the created ontology can be a struggle. This, therefore, means that this approach is less feasible for a wide degree of applications.

The aggregation of ontology domain knowledge into the recommendation process is one of the solutions that can overcome the limitations of conventional recommender systems. Ontology-based (OB) recommenders are knowledge-based systems that use an ontology to represent knowledge about the items and users in the recommendation process. In education, ontology-based recommender systems employ ontological knowledge regarding the students and the courses to map a student to relevant university courses which meet their individual needs. Ontologies play an important role in knowledge representation as well as in knowledge sharing and are reused in these systems. Previous studies have shown that aggregation of ontology domain knowledge regarding the learner and the learning resources improves the accuracy and quality of recommendations as well as alleviating other drawbacks associated with conventional recommendation techniques such as information overloading and cold-start problems (Adomavicius and Tuzhilin 2005; Zhao et al. 2015b).

The automated approach can construct ontology without the input of user mediation. However, several problems exist in this unsupervised method such as inconsistency and, in particular, the appropriate handling of missing information of concepts and their relationships deserve mention. In addition, the factor of inconsistency plays a crucial role in diminishing the effectiveness of this method. Therefore, the construction of high quality and efficient ontology remains an open research problem. To overcome the limitations in the available construction process, an automated ontology approach is proposed.

The ontology model includes the construction of dynamic ontologies for the user and the items that map these ontologies in order to gain a comprehensive knowledge for the recommendations. After building the ontologies and mapping them, the result will be used as an input in the recommender engine. In the proposed approach, ontologies are used to model knowledge regarding the course content (the course profile), knowledge about the user (the student profile) and domain knowledge (the taxonomy of the domain being learned). Within the domain of knowledge aspect, the term ontology refers to both the formal and explicit descriptions of the domain concepts (Ibrahim et al., 2017). These are frequently considered to be a set of entities, relations, functions, instances and axioms. By enabling the users or contents to share a common understanding of the knowledge structure, ontologies provide applications

with the ability to interpret the context of the student profiles and the course content features based on their semantics.

In addition, the hierarchical structure of the ontologies allows developers to reuse the domain ontologies (for example, in computer science and programming language) (Singto & Mingkhwan, 2013) in order to describe the learning fields and to build a practical model without the need to begin again from scratch. The present work has constructed three ontologies. Firstly, the course ontology, secondly the student ontology and thirdly, the job ontology. The protégé tool has been used to evaluate the ontologies with the hierarchical mapping between the ontology classes that are used to compute the similarity between them. Knowledge, represented by the ontologies, has been combined into one single ontology. The ontology model thus created significantly helps to reduce information overloading.

In order to understand how to find semantic similarity between the items, two important modules need to be explained in this section; firstly, the automatic dynamic ontology construction module (including item ontology and user ontology) and the ontology mapping module for which more details will be provided in chapter 4.

### **3.3.2.1 Dynamic ontology construction**

The difference between static and dynamic ontology is that the dynamic depends on certain parameters changing that can be considered globally to be situations. Static and dynamic ontologies are suitable examples of the static and dynamic concepts of classical physics (Middleton et al., 2009). There are generally several ways to make a given static ontology become a dynamic one; it simply depends on what is to be defined as being changing objects. However, ontologies developed by static approaches consist of terms that are limited in their knowledge base due to a lack of updating. A dynamic ontology-based model is proposed to classify the extracted terms and to build a knowledge base for a specific domain. It is a challenge to obtain a well classified corpus. Even if a corpus is available, it may be classified incorrectly due to fewer terms being classified because of the limited and static nature of the classifiers. To overcome this, the use of an ontology-based model is proposed in order to classify the terms and prepare the knowledge base.

Ontology is a data model that characterises knowledge about a set of classes or concepts and the relationships between them (Antoniou & Van Harmelen, 2008). The classes define the types

of attributes or properties that are common to individual objects within the class. The following modules explain our proposed dynamic ontology model: Document Analysis, Ontology Construction as shown in Fig.3.5.

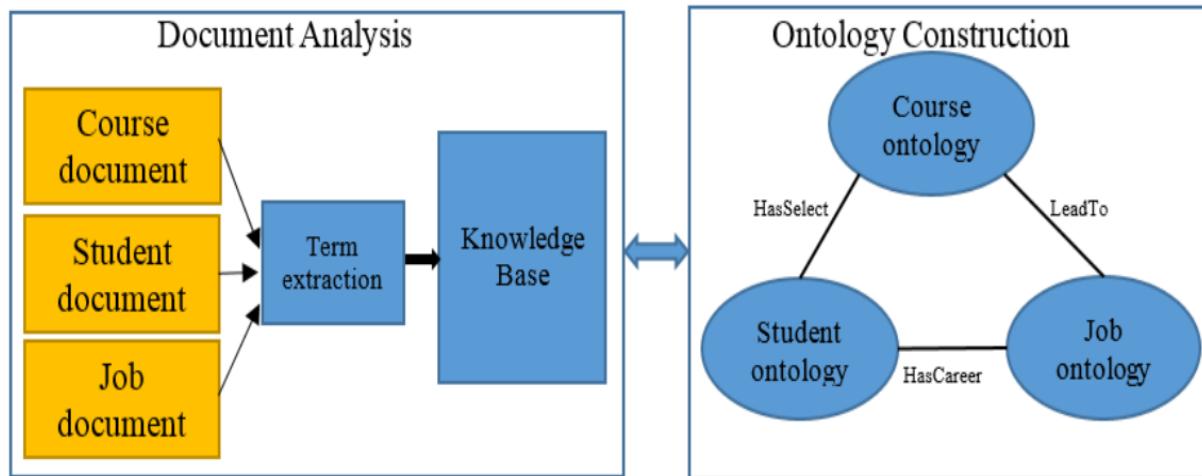


Figure 3.5 Dynamic Ontology Construction

There are many existing methods of constructing ontologies available. In the present work, the “Ontology Development 101” approach developed by Natalya Noy and Deborah McGuinness (X. Zhou et al., 2004) is followed. The language used to write the ontology is OWL 2 Web Ontology Language (Kadima & Malek, 2010) and the protégé tool Version 5.2 (Jambhulkar & Karale, 2017) has been used to build the model. In order to construct this ontology, the following steps have been considered:

1. Determine the domain and scope of the ontology

In this thesis, higher education has been determined as the domain, and master's courses in Computing and Business Management have been determined as the scope of the ontology.

2. Take into account reusing existing ontology

In education, many ontologies were found that model this aspect of the domain. However, no ontology was found that could be reused to serve our intended purpose. Despite this, current ontologies have been used as a guideline to model the common concepts of the new ontology.

3. Enumerate the domain terms

The ontology is defined as a taxonomy that helps to describe different aspects of the domain, such as the student, course and career. Some concepts are further divided into subclasses that would improve the classification of the instances of these classes.

#### 4. Determine the classes and the class hierarchy

The classes are defined as a group of individuals or instances that represent a class where all of the members share the same concepts. When the classes are ordered hierarchically, this is termed a taxonomy. Inference engines use hierarchies to denote inheritance relationships. Classes are defined by following the combination development process, which is a combination of both bottom-to-top and top-to-bottom approaches. When this approach is followed, the important terms are first defined, and then generalisation and specialisation takes place.

#### 5. Define the relationships between classes

The relationship that exists between class members in an ontology is termed the properties. There are two types of properties: object and data properties. Object properties represent the binary relations that exist between members of the classes, such as the relationship between a student and the courses. Here, a property called *HasSelected* has been defined which is used to represent this relationship. Data properties link an individual to a data literal, such as a student's ID, and it was found that, by analysing users belonging to a particular profile, they have a similar interest in course ontology. Thus attributes such as *offereCourse*, *HasCareer*, etc. can help to decide initial recommendations for a user according to his/her profile. In addition, this work on the recommendation of courses has focused mainly based on CBF and the attributes in the course vector such as course title, main subject of course and location. The user nodes in the user profile ontology are linked to course attributes in the course ontology using a *hasFeildOfStudy*, *HasLocation* relations. The course ontology is linked with job ontology using a *LeadTo* relation.

#### 3.3.2.2 Ontology Mapping Module

After constructing all the local ontologies in the framework, it is essential to discover the links between these ontologies. Mapping ontologies will help to obtain a comprehensive knowledge to answer queries asked by users. However, as each domain uses its own set of ontologies, an interoperability issue arises when exchanging information among these domains. To overcome this interoperability issue, an ontology mapping module (OMM) proposed a new mapping algorithm to establish a mapping between ontologies in the framework. The ontology mapping algorithm focuses mainly on improving the efficiency and accuracy of the mapping process whilst also addressing other issues regarding the declarative and expressivity of the mapping

representation. The mapping between the two ontologies is performed at two levels. The first level maps the concepts between them while the second level matches the properties for a given set of mapped concepts. The mapping process has been discussed in more details in chapter 4.

The mapping process includes several steps, starting with two ontologies which are going to be mapped as its input. The derivation of ontology mappings takes place in search of candidate mappings. The similarity computation determines the similarity values of candidate mappings. Hypotheses are then generated using a rule base. This rule base contains a set of deductive rules which may be enriched with new rules proposed by domain experts. The “best” similarity hypothesis is selected. Each step can be repeated for multiple rounds and exchanges messages with the previous step if necessary.

In OPCR, the recommender systems utilise domain ontologies to enhance personalisation because, in CF clustering, the interests of the user are modelled more effectively and accurately through ontologies and the application of a domain-based inference method. OPCR will use an ontology method to improve accuracy and to enhance personalisation in the CF aspect of the hybrid recommendation system using ontology and content of items. Ultimately, content-based features and ontology can be considered for the improvement of personalised recommendation and accuracy in the CF aspect by combining memory-based and model-based techniques.

### 3.3.3 Recommendation Component

After constructing the ontology models, in this section, the recommender engine is now introduced. OPCR used a hybrid method that combined the CBF and CF filtering approaches with supporting ontology model mapping, and this is the core component of the framework. In chapter 4, how each element of the hybrid approach works is explained in detail. Furthermore, the recommendation component has the following modules:

#### 3.3.3.1 Recommendation Engine

The Recommendation Engine (RE) is a tool that contains one or more different types of an algorithm that have been combined to recommend items to users based on user preferences. In order to provide a personalised recommendation to users, a series of stages will be implemented, and a different score will be calculated at each stage according to the weighting of that stage. The details of each stage will be discussed in different scenarios in chapter 4.

The OPCR recommends courses to students according to their profiles as well as in relation to similar users with whom they share a comparable academic background and who, when they used the system, had highly rated the courses. The user is required to provide basic personal information and academic information such as field of study, main subject of the course, interest area and the type of skills that he/she has. In the case of recommending a course at UK universities, there are certain important factors that need to be considered by the user such as the range of the course fee, the range of the university ranking and also the range of the degree of NSS score. After all this information has been provided by the user, the recommendation algorithms will then be implemented according to the CBF and CF filtering approach in order to provide the user with relevant recommendations. However, in a situation where the user wishes to search for a course using OPCR without first providing user profile information, the find requester will search for the course in the database according to a similarity between the given keywords and the course titles using TF-IDF technique. The keywords relevancy is calculated by multiplying term frequency with the inverse document frequency.

### 3.4 User Interface Component

The User Interface (UI) component facilitates the interaction between users and the system of a service provider. The users' interaction is performed through user registration, user login, user on-demand requests and corresponding recommendations and user feedback. HTML was used for the system interface. The UI component consists of two main modules: user module and administrator module. The user module contains several activities such as registration onto the system, login to the system, conducting a general course search, adding ontology concepts, undertaking a personal search based on ontology and displaying a list of recommendations to obtain the user rating for the recommendation items. Through the UI component, a user can perform the following actions:

- Login to the system every time
- Create a user profile through the registration page
- Search courses and create a personalised recommendation list
- Add, update and delete items from the user profile and rate recommendation list items
- Provide feedback regarding the received recommendation

The administrator module has many activities that make the framework more flexible and able to adjust, delete, add and manage ontology models classes without the need to access the back-

end of the framework. The administrator module also allows adjustment to the weights of the algorithms in the framework by using an algorithm weight adjustment page. The module interactive interface between the system and active users. It will be generated based on the respective user's information including the user's demographic information, personal interests and requirements within a given domain of education. Making a framework that enables a user to modify their personal information dynamically can lead to a reduction in the time taken to create a new ontology.

### 3.5 Summary

This chapter has presented the details of the proposed ontology based personalised course recommendation framework (OPCR). OPCR is made up of different components that are important for the framework to work effectively. The proposed framework supports the development of personalised course recommendation systems. OPRCourse is designed to provide course recommendations to students based on the ontology concepts similarity between the course profile and the student profile. It takes into account the personal information and academic information of the users in the recommendation process. Furthermore, an ontology mapping module is proposed in the OPCR framework to integrate information from multiple sources and to map this into a single unified module in order to obtain a comprehensive knowledge with which to answer users' queries. The aggregation of ontology domain knowledge into the recommendation process is one of the solutions that can overcome the limitations of conventional recommender systems. Ontology-based (OB) recommenders systems are knowledge-based and use ontology to represent knowledge about the items and the users in the recommendation process. In addition, user profiling that is based on ontology, item ontology and the semantic similarity between two ontologies is used to overcome the new user problem

Moreover, OPCR is a novel, personalised, adaptive dynamic hybrid recommender framework which supports the solution to the information overloading and cold start user problems which pertains to the difficulty of providing high quality recommendation to new users. OPCR supports the representation, indexing, sharing and delivery of context information and provides modular components that are common across applications.

## **CHAPTER 4 ONTOLOGY MODEL AND RECOMMENDATION ENGINE ALGORITHMS**

### **4.1 Introduction**

The rapid increase of information available on the WWW in different domains has caused a problem of information overloading. As explained in chapter 2, the education domain is one of the domains which have been influenced by this problem. Finding the appropriate education facility or relevant course has become a challenging task for most students. Before they enrol on a relevant university course, a student needs to be certain that the choice most successfully matches their individual needs. Recommender systems represent a promising approach by which to tackle the problem of information overloading. While university courses may feature similar course concepts within the same field of study, the various courses may lead to different type of career paths, therefore, the ontology model has been used to solve the semantic similarity problem.

There are many factors that influence students when they seeking to make a decision regarding the selection of a university course as mentioned in chapter 2 and information regarding these factors is available in different formats and from different sources. The ontology model allows information from multiple sources to be automatically integrated in order to obtain comprehensive knowledge with which to respond to the queries raised by users/students. Furthermore, modelling the information at the semantic level is one of the main goals of utilising ontologies.

This chapter is dedicated to describing in detail both the construction of the ontology model and the recommender engine algorithms that have been used in OPCRa and how this approach can tackle information overloading and cold start problems experienced by a new user. In the ontology model section, a new approach is introduced to automatically generate ontologies for both the item (course/job domain) and the user (student domain). In addition, a new mapping algorithm is proposed to map the similar concepts of the domain ontologies to obtain a comprehensive knowledge of recommendations items. The mapped information is used as an input in the recommender engine. OPCRa attempts to reduce the confusion experienced by the user when attempting to retrieve what he/she wants from the massive number of items (for a certain specified domain).

The second part of this chapter will discuss in depth the algorithms which have been proposed for the recommendation engine to produce recommendations for users. A hybrid filtering approach is used in OPCRa which combines the CBF filtering approach and the CF filtering approach and utilises ontology to enhance the recommender algorithms. Using ontology in the recommender system helped to increase the accuracy of user similarity. In the following subsections, more details will be discussed regarding the ontology model and the filtering algorithms in OPCRa.

## 4.2 Ontology model

The aggregation of ontology domain knowledge into the recommendation process is one of the solutions that can overcome the limitations of conventional recommender systems. Ontology-based (OB) recommenders are knowledge-based systems which use an ontology to represent knowledge about the items and users in the recommendation process. In education, ontology-based recommender systems use ontology knowledge about the students and the course resources in order to map a student to relevant university courses which meet their individual needs.

Ontologies are used in OPCRa to model knowledge about the course content (the course profile), job content (the job profile), knowledge about the user (the student profile) and domain knowledge (the taxonomy of the domain being learned). Within the domain of knowledge representation, the term ontology refers to both the formal and explicit descriptions of the domain concepts (Ibrahim et al., 2018). These are frequently perceived as a set of entities, relations, functions, instances and axioms. By enabling the users or contents to share a common understanding of the knowledge structure, ontologies give applications the ability to interpret the context of the student profiles and course content features based on their semantics. In addition, the hierarchical structure of the ontologies allows developers to reuse the domain ontologies, for example, in computer science and programming language (Singto & Mingkhwan, 2013), in order to describe the learning fields and to build a practical model without the necessity of starting from scratch.

In this thesis, three ontologies have been constructed to validate the proposed approach. Firstly, the course ontology, secondly the student ontology and thirdly, the job ontology. Fig.4.1 shows

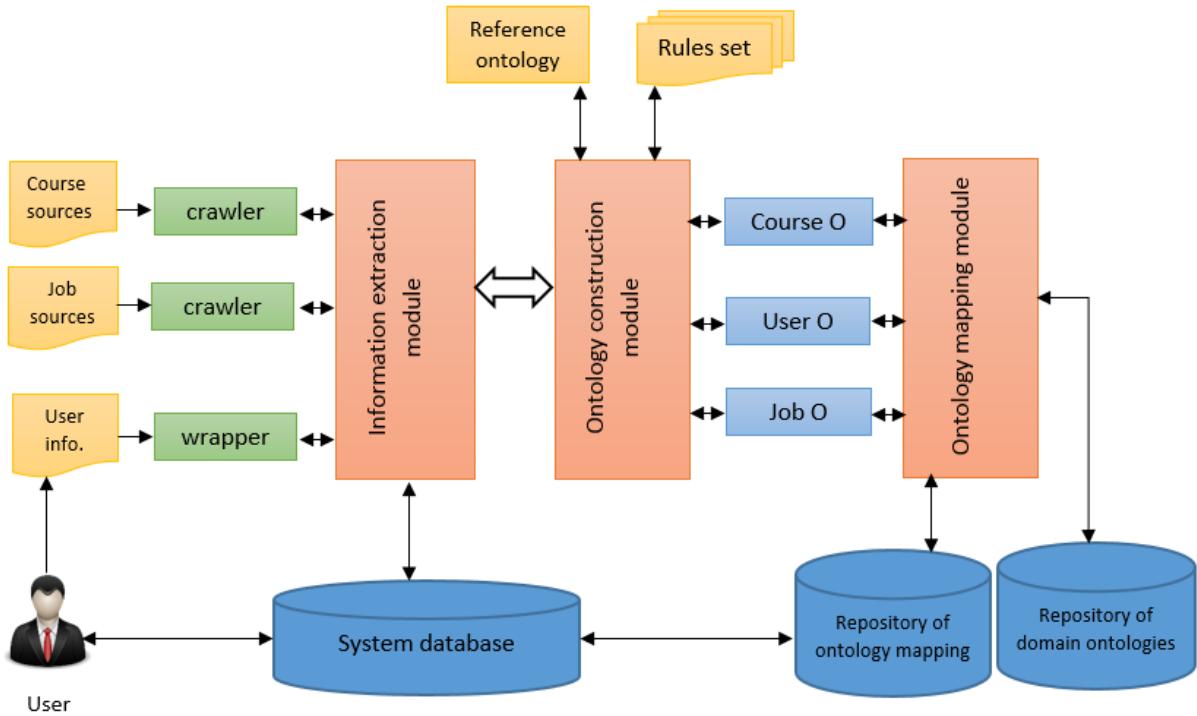


Figure 4.1 Data flow and main related module in ontology model

the data flow and the related modules developed to build the ontology model. The protégé tool has been used to evaluate the ontologies with the hierarchical mapping between the ontology classes used to compute the similarity between them. Knowledge, represented by the ontologies, has been combined into one single ontology. The ontology model that has been created significantly helps to reduce information overloading.

#### 4.2.1 Ontology Construction Module

All the useful information collected from multiple sources is presented as entities, and their relations are all described by the relationship in the relational database as mentioned in chapter 3 section (3.3.1). These relationships correspond to concepts in the ontology. The information is represented in tables in the database. Three main tables have been presented in the module which are a course information table, a student information table and a job information table. When generic domain ontologies and domain ontologies are modelled, the ontologists should select the top-level ontologies to be reused. And then the application domain ontologies are built on top of them(Fernandez-Lopez & Corcho, 2010). In ontology construction module Zaemmoruchi and Ghomari approach has been used to build reference ontologies(Zemmouch & Ghomari, 2013) . Reference ontology is generally constructed using the concepts of other ontologies as inputs to an analysis and subsequent synthesis. So, the commonly used approach

in constructing the reference ontology is to survey all concepts from all established ontologies and analyze similarities and differences, in order to arrange concepts into a coherent representation of ontology domain knowledge.

Reference ontologies are primarily designed as extensions or specializations of high-level ontologies that take a global view of multiple domains of reality, and do so in accordance with principles of ontology science. The same principles and set of rules have been used to transfer the information from each table to a local ontology. According to (Lei Zhang & Li, 2011), five rules have been identified by which to construct an ontology from a relational database as follows:

**Rule 1:** Read the information from tables in the database, map the relationships directly into ontology concepts, defined as class  $C_i$ . Map a data table into a concept. The table name acts as the concept name. The table attributes act as the concept properties.

**Rule 2:** For the relationship,  $R_i$  in a relational database, Supposed  $P_i = PKey(R_i)$ ,  $F_i = FKey(R_i)$ , if  $F_i(R_i) \subseteq P_i(R_i)$ , then the attributes of foreign keys are removed from the properties of the concept. That is, the attributes of foreign keys are not considered as the properties of the concept to map.

**Rule 3:** For the relationship in the relational database, if  $FK(R_i) \neq 0$ ,  $PK(R_i) \neq 0$ ,  $FK(R_i) = PK(R_i)$ , then the table is a bridge table. Two object properties (owl: ObjectProperty) of ontology are created from this. Both properties are reciprocal properties. Their domain and range are the two ontology concepts corresponding to the relationships which are referenced respectively.

**Rule 4:** For the inheritance relations  $R$  and  $sub R$  in a relational database, these can be mapped into ontology object properties (owl: ObjectProperty) directly. The OWL class which corresponds to the table  $sub R$  is declared as a subclass. The corresponding class of table  $R$  is seen as a parent class.  $Sub C$  is a subclass of  $C$ .

**Rule 5:** For the relationships  $R_i$  and  $R_j$  in a relational database, if  $F_i(R_i)$  is the foreign key attributes of  $R_i$ , and  $F_i(R_i)$  is referenced by  $R_j$ . Moreover, they cannot meet rule 3 and rule 4. Then, object property (owl: ObjectProperty), “HAS\_A” is added to the foreign key attributes. The domain and range are  $C_i, C_j$  respectively.

The ontology construction process begins by extracting information and a relational model from the database and establishing a relation metadata model. Based on the analysis of the relational database model, it transfers the relational database into the ontology model using the

rules defined above. Ontology information is thus extracted from a relational database by mapping the database data into ontology instances according to the data conversion steps from the relational database to OWL ontology. Fig.4.2 shows an example of course ontology construction from the course table.

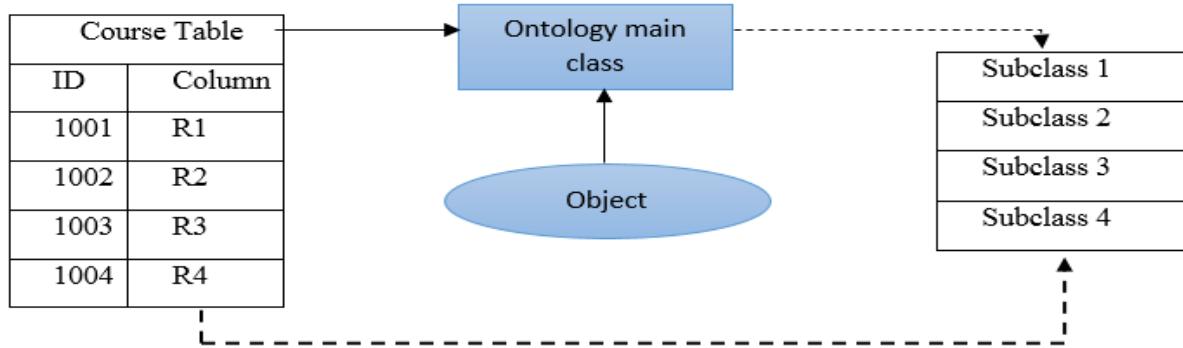


Figure 4.2 Construct course ontology from course table

#### 4.2.2 Course Ontology

Identifying different attributes is necessary for course profiling (Lee, 2011). In order to construct a course ontology, the factors that most influence a student when they make a decision about the choice of a university course need to be identified. These factors then form the main classes of the ontology. Students at the University of Portsmouth were surveyed to ascertain the most important factors that had influenced their choice of university course. More than 200 students participated in this survey. They were given 20 factors that influenced their decision when choosing a university course programme and were then asked to rank these factors on a scale of 1-10. The 20 factors were classified into six categories, and the scores and standard deviations for each category were computed. The results are summarised in Table 4.1.

Factors and key constituent element	Mean
Course information( Field of study, Courses , major subjects , course structure)	7.8
Course Fee	7.5
NNS score	7.4
Prominence (institutional reputation )	6.4
Location ( institutional location)	6.9
Career	7.9

Table 4.1 Factors and keys constituent elements for selecting university courses 60

The title and fees of a course programme and the location and prominence of the university were all factors that appeared to be the most important when students determined their choice of university for higher education (HE) study. The following points can be noted:

- ❖ Taking 5.5 as the midpoint on a ten-point Likert scale, three of the seven factors had a mean score that was lower than this midpoint. It can be assumed therefore that the promotion, people and prospectus elements do not have a significant influence on the choices that students make regarding where to study for their higher education.
- ❖ Among the elements included in the programme factors, both the field of study and details regarding the course information appear to exert the most considerable influence on the / choice of university course programme by students.
- ❖ The factor that was uppermost in the' decision-making frameworks of the students was the issue of fees which had the greatest impact on university choice and the type of career that could be achieved following completion of the course.
- ❖ It was found that issues of institutional prominence maintain a fairly high profile in students' decision-making. The overall reputation of the institution and the national student survey score (NSS) of teaching students are both significant.

The course attributes are considered when extracting the course profile including the essential information, course information, as well as information regarding fees and university rankings and the NSS score for the university. This information is used for knowledge discovery at a later stage of the user profiling process. In Fig. 4.3 the main classes and subclasses of the course ontology with instances are shown.

The course profile attributes will match the user profile features through ontology mapping. Each class of the course profile will be a map to the equivalent class in the user profile. Ontology reference is used to identify the equivalent classes in both the course profile and the user profile. The protégé tool was used for the construction and evaluation of the ontology model. Fig.4.4 shows the graphical representation of the course ontology in the protégé environment.

Fig. 4.4 shows the concepts hierarchy in the course ontology features parsed at different levels. Some of the features contain a number of other aspects which are related to the main feature. For example, *course\_information*, as the main feature in the course profile, takes the position

at level 1 and includes four features which are *MajorSubject*, *FieldOfStudy*, *Course\_title* and *MainSubject*. Each of these features is assigned by weight in the scoring function according to the degree of importance of the feature. In the example of the *Course\_information*, the features of *course\_title* and *MajorSubject* are given high importance in order to calculate the similarity between two courses. Therefore, when the user conducts a search, the ontology hierarchy will help the system to find the user's interests by checking the concept in the user profile feature and calculating the similarity in the *course\_information* feature. For example, if the feature *FieldOfstudy* is Information Technology and *MainSubject* is computer sciences in the course profile, the system will calculate the similarity only with the courses under computer sciences instead of looping in the entire database. Thus, using the ontology model leads to an improvement of the performance of the recommender system.

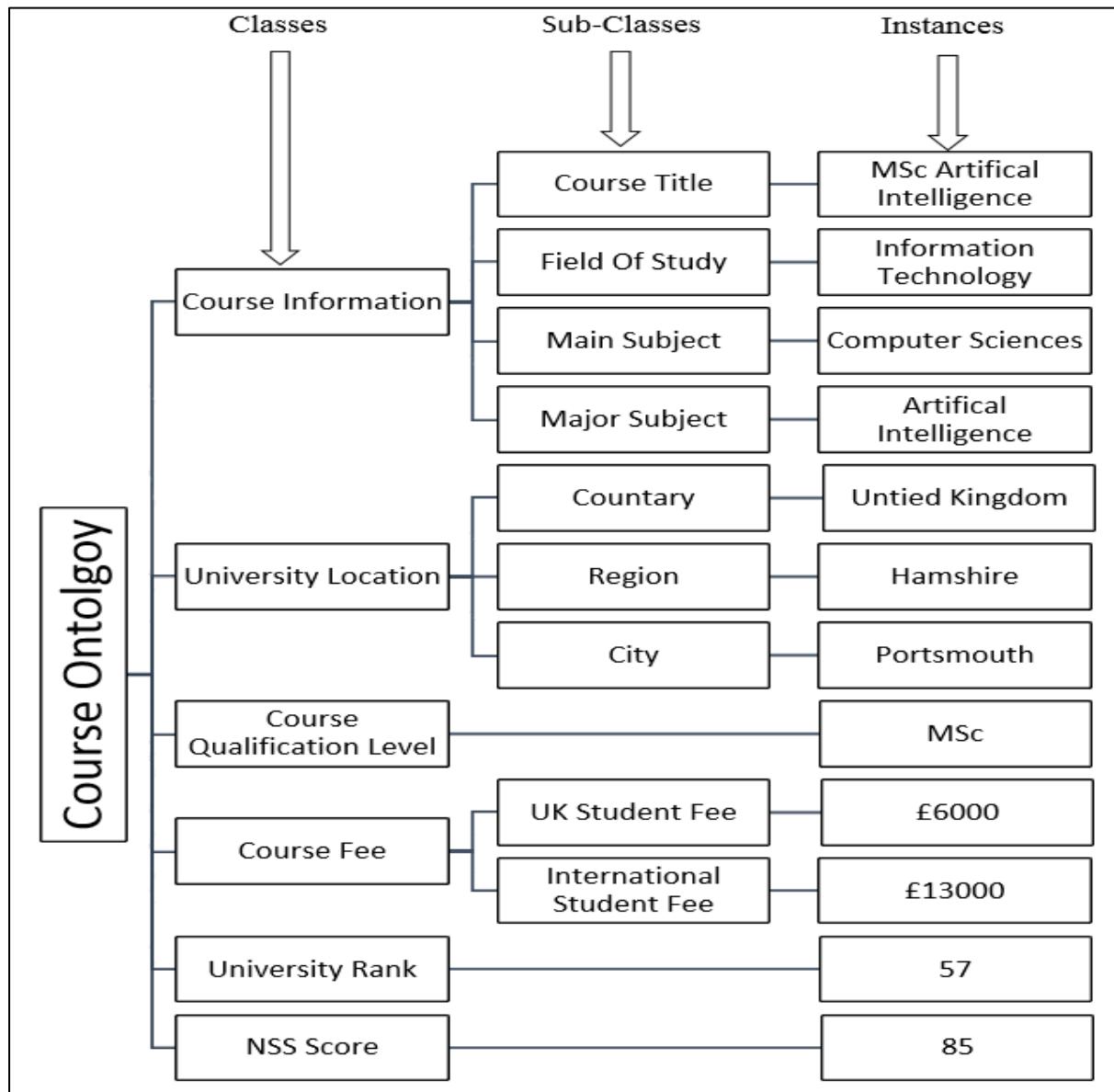


Figure 4.3 Course ontology structure

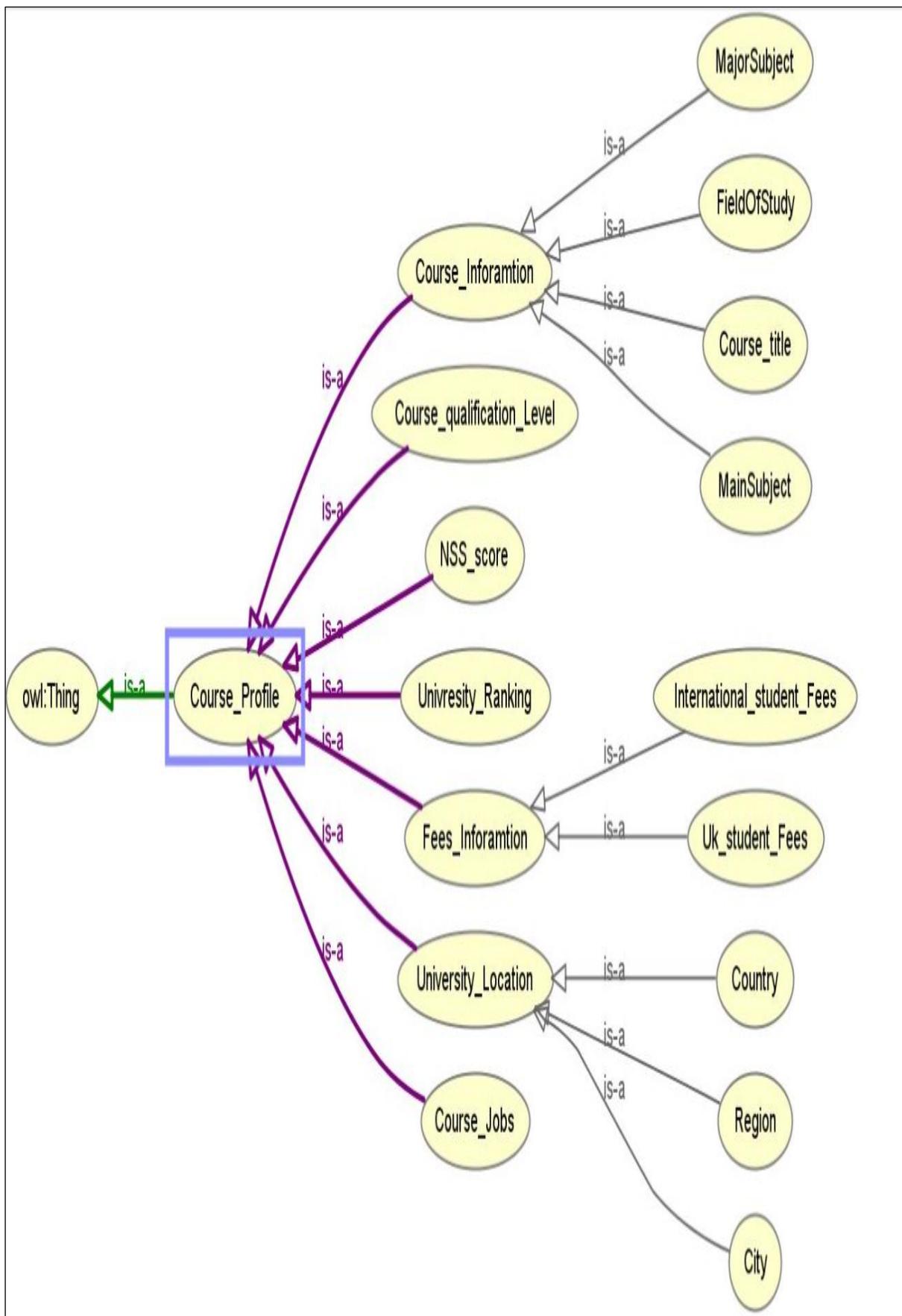


Figure 4.4 Graphical representation of the course ontology in the protégé environment

### 4.2.3 Student Ontology

Firstly, the student profile needs to be modelled prior to recommending an appropriate course. The user profile consists of two main parts. The first part is the personal and educational attributes of the user and the second is the user's rating of the previously recommended course. The personal and educational attributes include the user's individual personal information as well as education and background information such as their hometown, gender, the field of study, main subject, major subject, interest area, technical and non-technical skills. In Fig. 4.5 and Fig. 4.6, the graphical representation of student profile ontology in the protégé environment is shown. A student profile can be defined as:

$$U = \{ a_1, a_2, \dots, a_n \} \quad (4.1)$$

Where  $U$  is the user/student,  $a_i$  represents the user's  $i$ th attribute.

If a student has obtained an offer from the system in the past and rated the courses, that student can be further defined as:

$$U_r = \{ u, r \} = \{ a_1, a_2, \dots, a_n, r_n \} \quad (4.2)$$

Here,  $U_r$  is the user that received a recommendation for the courses from the system and has rated the courses.

Furthermore, in order to make a satisfactory recommendation, it is important to ensure that the characteristics of the recommended activities match the interests of the user. The course ontology is created for all the courses that are to be recommended to the user/student. The system recommends several courses in the streams of arts, information technology, science, social science, management, commerce, engineering, education and law. The student obtains a recommendation for any course depending on their eligibility, i.e. if the student has a graduate degree, the system can recommend any post-graduate course. If the student has a postgraduate degree, then either a research course or PhD can be selected, depending on the faculty. The proposed approach conducts an entrance test as an eligibility criterion for admission onto undergraduate and postgraduate engineering courses.



Figure 4. 5 Graphical Representation of the Student Ontology

In the proposed system, there are three ontologies: course ontology, student profile ontology and jobs ontology. There are three aspects of the local ontology construction process. These are unstructured text documents from structured relational data sources and semi-structured data sources. Unstructured text documents include four processes: data pre-processing, concept clustering, context extraction and local ontology construction. For more information regarding local ontology construction from the unstructured text, see (Ibrahim et al., 2018) .

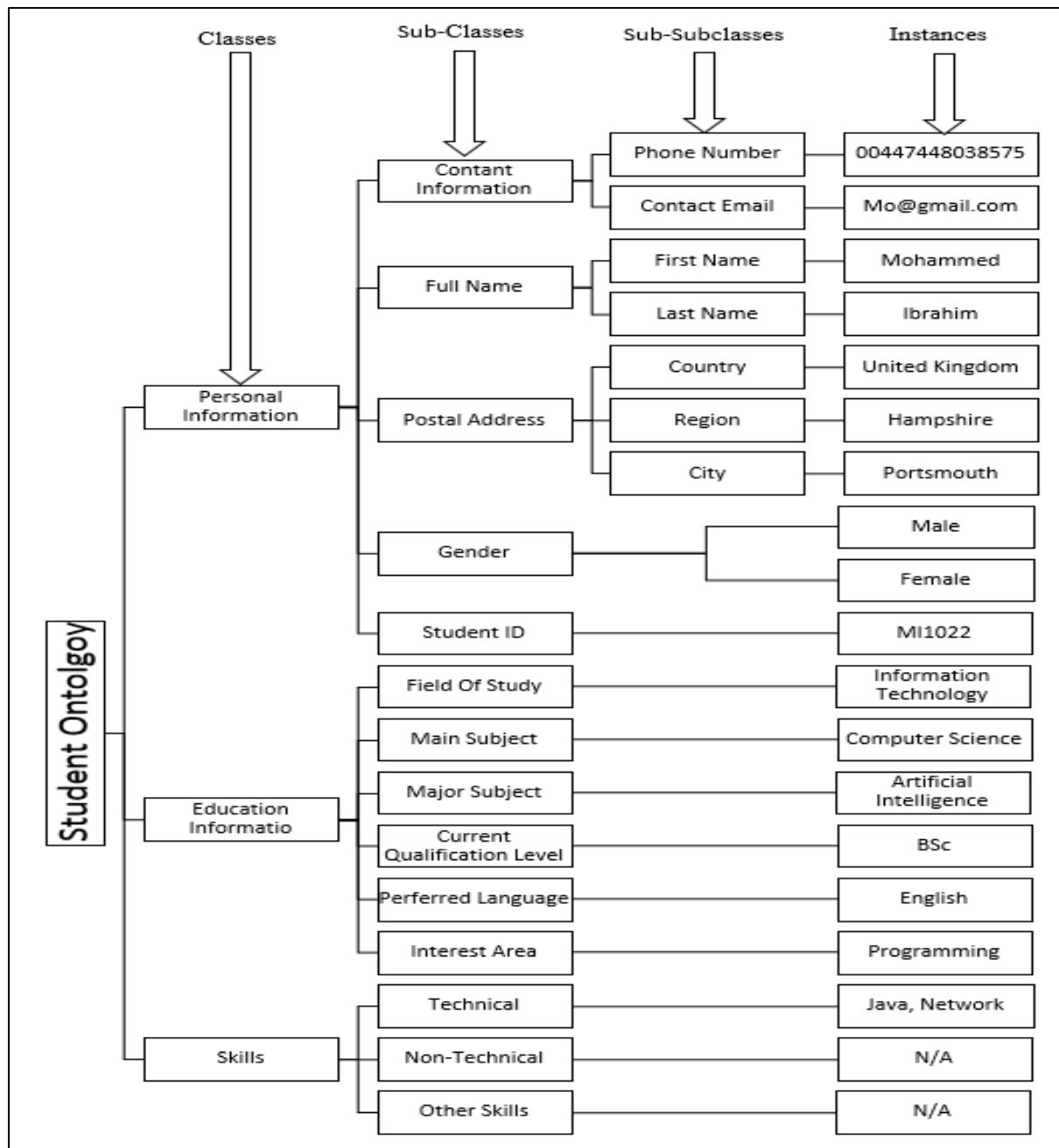


Figure 4.6 Student ontology structure

#### 4.2.4 Job Ontology

The intended future career of a student is an essential factor that can influence decision making when selecting a university course (Farzan & Brusilovsky, 2006). Constructing a job ontology is vital if a student is to understand the attributes of the planned employment role and career path. This information is extracted from a job website, such as Indeed.com. Job attributes include such information as job title, job description, job salary, job location and the required educational qualifications, as shown in Fig. 4.7. A graphical representation of the job ontology in protégé environment is shown in Fig. 4.8

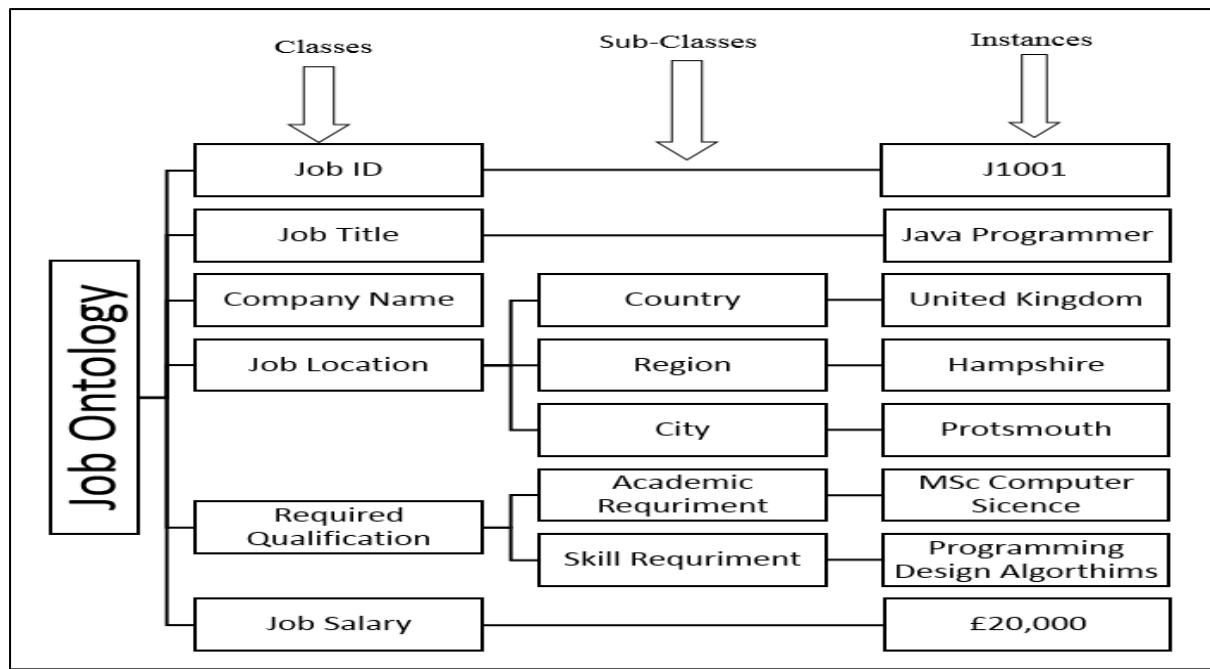


Figure 4.7 Job ontology structure

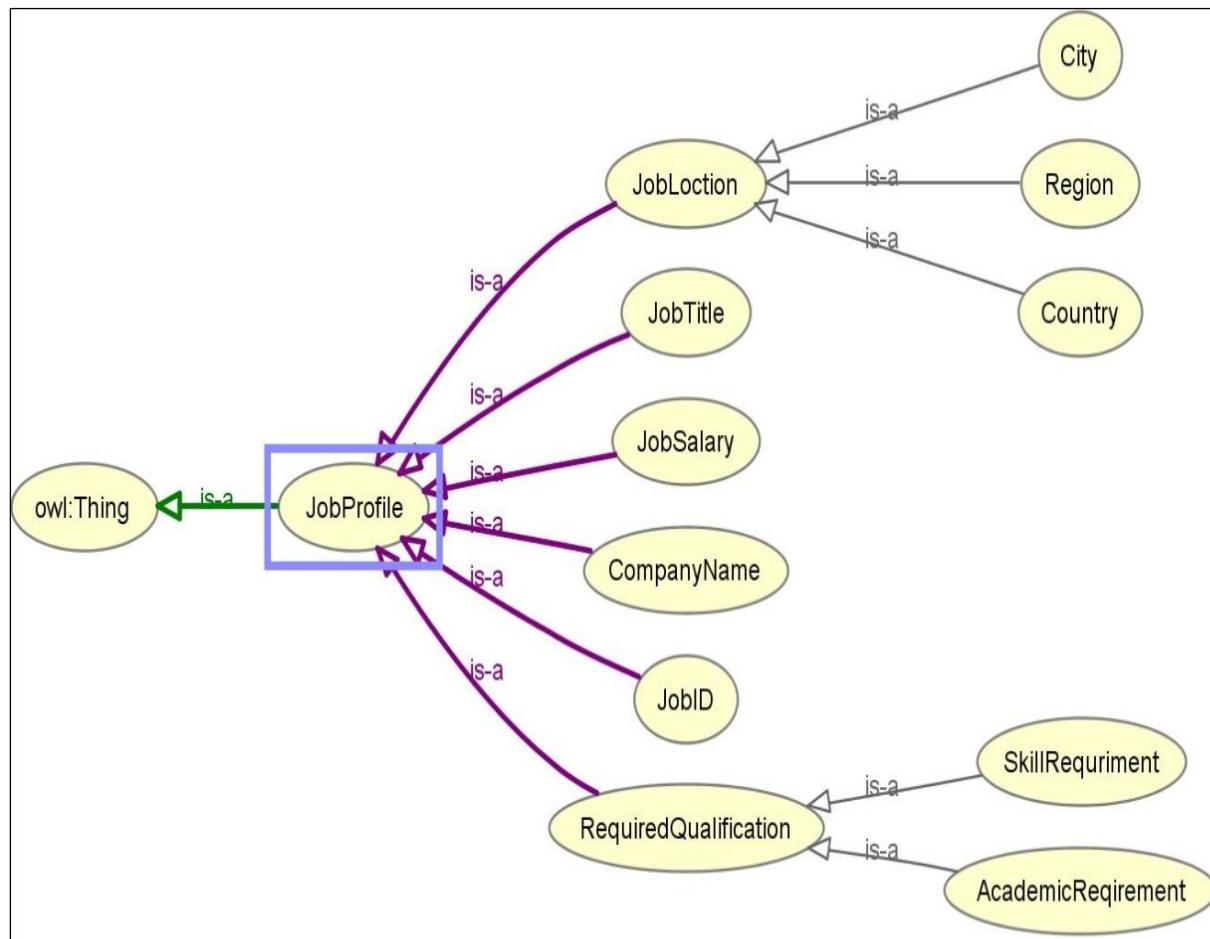


Figure 4.8 Graphical representation of the job ontology

### 4.3 Ontology Mapping

After constructing all the local ontologies, it is essential to discover the links between these ontologies. Mapping ontologies will help to obtain a comprehensive knowledge with which to answer users' queries. However, as each domain uses its own set of ontologies, an interoperability issue arises when exchanging information among these domains. To overcome this interoperability issue, an ontology mapping algorithm is proposed to establish a mapping between the ontologies. Ontology mapping algorithm focuses on improving the efficiency and accuracy of the mapping process whilst also addressing issues regarding the declarative and expressivity of the mapping representation. The mapping between the two ontologies is performed at two levels. The first level maps the concepts between them and the second level matches the properties for a given set of mapped concepts. The mapping process is as follows:

1. Transform the local ontologies to the system uniform representation model; local ontologies are often different in formalism, mechanism and language. In order to compare and find similar concepts between the ontologies, the system needs to represent all ontologies in a uniform model.
2. Choice of one domain ontology related to the scope of local ontologies. A domain ontology for each domain area with which the system works needs to be created. For instance, if the system is designed for a course domain, then a course ontology should be created and modelled in the system uniform representation model and the same applies for the student domain and the job domain. This domain ontology should be complete and cover all the required elements and terms in the domain.
3. Determine similar concepts between local ontology  $O$  and domain ontology; the mapping algorithm compares local ontology  $O$  terms with domain ontology terms and discovers similar pair terms.
4. Determine similar terms between domain ontology  $O$  and local ontology  $O'$ ; the mapping algorithm compares domain ontology  $O$  terms with local ontology  $O'$  terms and discovers similar pair terms.
5. Map local ontology  $O$  terms to local ontology  $O'$  terms; the algorithm maps domain-based local ontology  $O$  to domain-based local ontology  $O'$  based on discovered similarity relations in third and fourth steps.

Given two ontologies O and O', mapping one ontology onto another means that, for each entity concept c in source ontology O, a corresponding concept c' needs to be found which has the same intended meaning as in the target ontology O'. The mapping process illustrated in Fig. 1 includes four main steps, starting with the two ontologies which are going to be mapped as its input. The derivation of ontology mappings takes place in a search of candidate mappings. The similarity computation determines similarity values of candidate mappings. Hypotheses are then generated using a rule base. This rule base contains a set of deductive rules which may be enriched with new rules proposed by domain experts. The “best” similarity hypothesis is selected. Each step can be repeated for multiple rounds and can exchange messages with a previous step if necessary. Table 4.2 shows the main classes for each ontology and the relation properties between ontologies. Fig.4.9 shows the concepts hierarchy of the mapping domain ontologies (course, student and job) using the protégé tool.

STUDENT	COURSE	JOB	Relation Properties
Education Background	Course field, here we need to make the main class and subclass ex. Engineering is a main class and CE, EE, is as sub class	Job Filed	HasFieldOfStudy, HasSubjectRoot, HasJobFiled
Preferable Language	Course Language	Job Required languages	HasPerferLanguge, HasLanguage, HasLanguage
Current Qualification ( Bsc , MSc Mphil, PhD)	Course Level	Job Required qualifications	HasCurrentQualification, HasLevel, HasRequiredQualificationLevel
Skills	Course perquisites	Job Required Skills	HasSkill, HasPerquisitesSkills, HasRequiredSkills
Interest Areas	Course main subject, course Major subject	Future job	HasInterestArea, HasMainSubject, HasMajorSubject
Postal Address	Course Location	Job Location	HasAddress

Table 4.2 The main classes of student, course and job ontologies with their relation properties

In OPCR, the recommender systems utilise domain ontologies to enhance personalisation because, in CF clustering, the user's interests are modelled more effectively and accurately through ontologies and by applying a domain-based inference method. OPCR will use an ontology method for improving accuracy and enhancing personalisation in the CF aspect of the hybrid recommendation system using ontology and content of items. Ultimately, content-based features and ontology can be considered for the improvement of personalised recommendation and accuracy in the CF aspect by combining memory-based and model-based techniques.

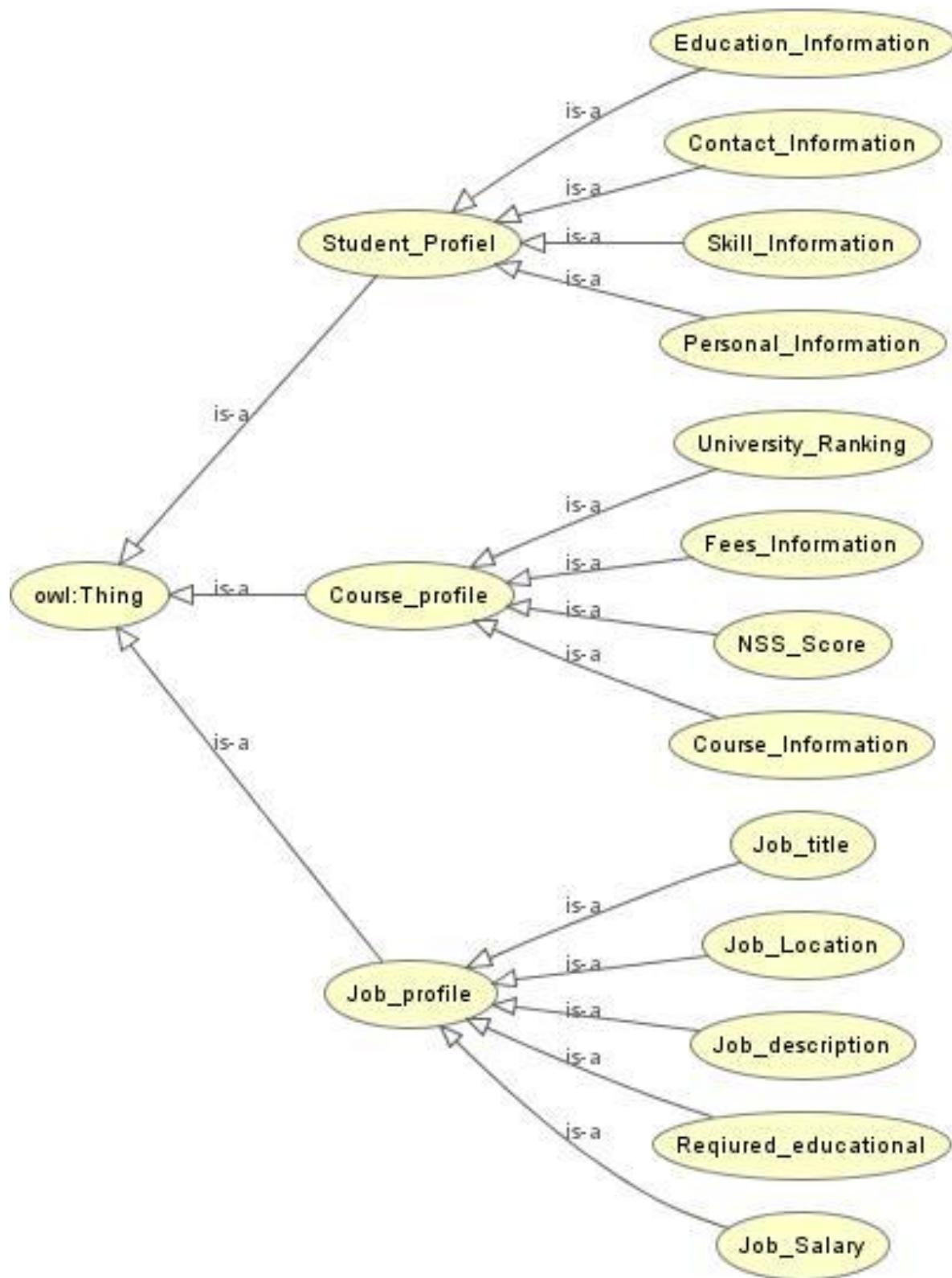


Figure 4.9 Concepts hierarchy for mapped ontologies

## 4.4 Recommendation algorithms

The primary target of this thesis is to overcome the problem of information overloading through the building of a hybrid recommender system that combines two main recommendation techniques, CBF and CF, with the utilisation of ontology. This combined model attempts to reduce the drawbacks and exploit the advantages of each approach. Also, using ontology in the CF part will improve the problems experienced by new users who have not rated any items. Ontology similarity (OS) will help to locate a similar user to the target new user.

After constructing the ontology models, in this section, the recommender engine is now discussed. A hybrid method which combined the CBF and CF filtering approach with supporting ontology model mapping was used. The proposed approach firstly applies content-based filtering by using the content of the user profile (personal features) where it will search for users who are similar to the features of an active user (personal profile) depending on the personal features and the impact ratio (weight) of these features. Then, if the active user is a new user or user who has already registered but has not had sufficient ratings, the personal similarity and the set of items genres of his/ her interest will be used to produce the list of recommendations. The elements will be selected from the domain of highly rated items of the nearest neighbours, and none of these is repeatedly recommended to such a user.

This step represents that used of Collaborative Filtering (CF) which depends on the opinions of similar users. However, if the active user has sufficient ratings, then the similarity between this user and other registered users will be calculated using the ontology similarity of the personal user information and memory-based collaborative filtering (the recommendation history rate of items). The final recommendation list is produced according to a combination of recommendations in both the CBF and CF methods. Each of these methods has different scores calculated at each stage depending on the weightings which were obtained through experiments where the best weightings were found to be as the follows: content-based filtering (CBF) -50%, collaborative-based filtering (CF) -30 %, university ranking -10 % and NSS score -10%. These percentage constants are obtained through trial and error in testing the performance of the system. These constants can be changed according to the application. The recommendations list will then be produced from the highly rated items of the nearest neighbours, and predictions on this recommendations list will be computed according to equation (2.5).

#### 4.4.1 Content Based Recommendation

As previously mentioned, CBF filtering is based on the similarities that exist between the items (courses) and the user's preferences. In order to calculate the similarity, a vector had to be generated for the features of both the item and the user. According to the course ontology model, the main classes are used as the features of the item vector. The features include the course title, the major subject of the course, the course fee and the location of the institution. A flexible weight has been adjusted for each of these features as 15%, 15%, 10%, 10%, respectively for each feature. An additional feature that was used in the CBF filtering to recommend the more relevant course was the reputation of the university and its NSS score. The weight assigned to each of these additional features in the final scoring function was 10% and 10%, respectively.

Different techniques have been used to calculate the similarity between the user profile and the course profile, according to the nature of the attributes in the course profile and the user profile. Hierarchy ontology similarity has been used for attributes, such as the course subject root and user preferred subject.

- Use cosine similarity to calculate the course title and major course subject, according to the formula (4.3)

$$Sim(If_a, If_b) = \frac{\overrightarrow{If_a} \cdot \overrightarrow{If_b}}{||\overrightarrow{If_a}|| \times ||\overrightarrow{If_b}||} \quad (4.3)$$

Where  $If_a$ ,  $If_b$  are item features of item  $a$ ,  $b$ .

- **Course fee similarity calculation:** The similarity between the university course fees and the user preferred fees has been calculated by using the following formula (4.4):

$$FS(U, C) = \frac{F_{umax} - F_c}{F_{max} - (F_{min} - 1)} \quad (4.4)$$

Where:

$FS(U, C)$  = the course fee similarity between the user preferred fee and the course fee for each university

$F_{umax}$  = the maximum university course fee that is expected from the user

$F_{min}$  = the minimum university course fee in the database

$F_c$  = the university course fee

- **Location similarity calculation:** The matching similarity has been used to compute the similarity between the user location and the location of the university providing the courses. In order to achieve more results, the user's city was also matched with the regions where the universities are situated. The United Kingdom has classified the cities, based on 12 regions, and each of the regions is formed of many cities. For example, the South East includes Portsmouth, Southampton and Kent amongst others.
- **University ranking similarity calculation:** The ranking attribute in the user query and course profile was calculated according to the formula (5.5)

$$RS(U, C) = \frac{R_{max} - R_c}{R_{max} - R_{min}} \quad (5.5)$$

Where:

$RS(U, C)$  = the university ranking similarity between the user preferred ranking and university ranking

$R_{max}$  = the maximum university ranking in the database

$R_{min}$  = the minimum university ranking in the database

$R_c$  = the ranking of the university providing the course

- **NSS score similarity calculation:** To find the similarity between the NSS score of the course and the NSS score that the user is satisfied with, the following formula (4.6) has been used:

$$NS(U, C) = \frac{U_N - (N_{min} - 1)}{N_{max} - (N_{min} - 1)} \quad (4.6)$$

Where:

$NS(U, C)$  = the NSS score similarity between the user and the course

$U_N$  = the user preferred NSS score

$N_{min}$  = the minimum NSS score in the database

$N_{max}$  = the maximum NSS score in the database

#### 4.4.1.1 Items representation design

In order to utilise the ontology similarities between the items and the users' interests, the items have to be listed as a set of topics that represents its relevant features. Each type of feature is included in a hierarchy of topics that belong to it, and this is the semantic information that the

recommender exploits. For example, if the course has a title related to artificial intelligence, the subject will be under the topic of computer sciences as the field of study.

With regard to the item's representation, it is assumed that the topics representing an item are always leaves of the feature hierarchy, that is, they are always specific topics and it is assumed that general topics cannot describe an item feature. This decision was considered in order to simplify the hierarchy-based similarity algorithm since, in this way, it is not possible that a given user's interest is more specific than the topic of the item.

As an outcome of the decoupled architectural design of the recommendation system to allow various application domains to be able to use it at the same time, the task of classifying the items with the particular topics of the domain is delegated to each application that will provide the correspondent pre-classified list of items in every recommendation request.

#### 4.4.2 Collaborative Based Recommendation

The previous section presented the way in which the CBF is able to calculate the similarity between the user profile and the item profile based on the available attributes in each profile vector. In this section, how the CF works and how using the ontology-enhanced CF performs to find the most similar users to the active user are explained. The most important aspect of the CF is how to measure the similarity between the active user and other users in the database. In addition, a new algorithm has been produced in order to enhance the KNN algorithm by using the ontology similarity called (OKNN). In the following sub-sections, each part will be presented in detail.

##### 4.4.2.1 User Similarity Calculation

The user profile vector consists of two parts; the first part is the user attributes, such as personal and academic information. The second part is the ratings that the user gives to the item in the CBF case. In the proposed work, a new method has been developed to calculate the similarity between the target user and other users in the database. The main idea is to use an ontology hierarchy similarity in the user profile and the user profile attributes. The proposed approach has ontology support from the user history similarity that enables it to calculate the similarity between the target user and other users in the system, according to the formula (4.7). The user similarity value range will be between (0, 1) and the weight for each part 50%.

$$US (U_a, U_n) = \text{ontology similarity} + \text{recommendation history similarity} \quad (4.7)$$

Where:

$US$  is a similarity between the target user  $U_a$  and the users in the system  $U_n$ . The system considers the levels of the ontology concepts in the user profile by classifying the ontology similarity to four levels. Moreover, the given weight for each level is based on its importance, as follows and as shown in Fig. 4.10, level 1 (major subject, main subject, the field of study), level 2 (interest area), level 3 (user location) and level 4 (user skills).

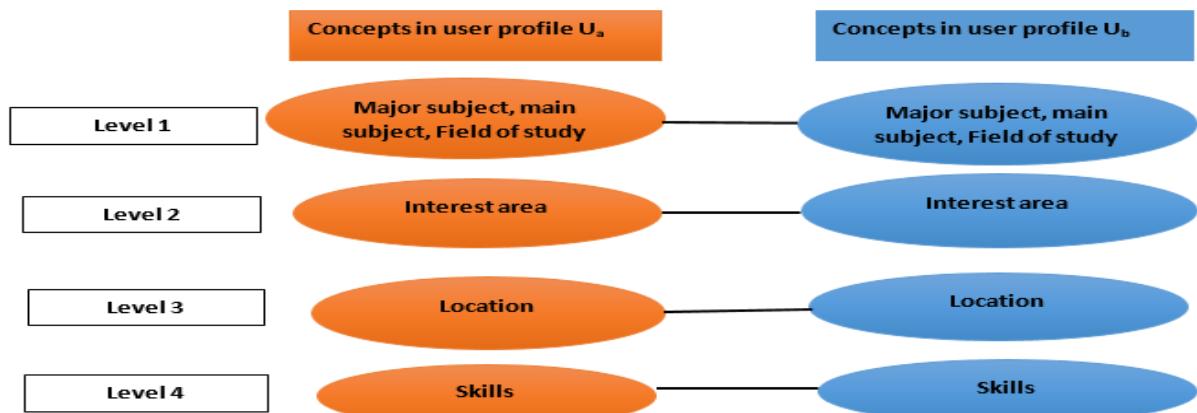


Figure 4.10 Hierarchical matching and matching parameters

To compute the similarity between each level of the ontology, the weight of each level needs to be adjusted based on the importance of the concepts in the levels. The importance of the concepts in the ontology level has been adjusted according to the results of the survey of postgraduate students at the University of Portsmouth. The results of the survey in section 4.2.2 showed that the concepts in Level 1 are more important when a user decides to choose a university course programme. The weight given to the levels is as follows; level 1 (30%), level 2 (10%), level 3 (5%) and level 4 (5%) respectively. For instance, if the  $U_a$  profile consists of these attributes: *artificial intelligence* as a major subject, *computer sciences* as a main subject, *information technology* as a field of study, *management* as an interesting area, *Portsmouth* as a location, *programming* as a skill, then user  $U_b$  profile has these attributes *computer programming* as a major subject, *computer sciences* as a main subject, *information technology* as a field of study, *management* as an interesting area, *Southampton* as a location, *programming* as a skill. The ontology similarity calculation between  $U_a$ ,  $U_b$  will be based on the Eq. (4.8):

$$OS(U_a, U_b) = \sum_{l=1}^n Lm \quad (4.8)$$

Where:

OS= Ontology similarity

N = number of levels in the ontology

Lm = level concept matching

$OS(U_a, U_b) = \text{level1} + \text{level2} + \text{level3} + \text{level4}$

$OS(U_a, U_b) = (0+ 0.1+0.05) + (0.1) + (0.05) + (0.05)$

$OS(U_a, U_b) = 0.35$

Moreover, after computing the ontology similarity, it will be necessary to obtain the recommendation history similarity between  $U_a$ ,  $U_b$ . In the proposed work, the recommendation history includes all the courses that have been rated by the user in the CBF case. Many algorithms have been applied to compute the similarity between the user recommendation histories. Cosine similarity is one of the algorithms that are most widely used in this area (Chang et al., 2016). The similarity between the users' recommendation histories has been computed according to Eq. (4.9) and Eq. (4.10), as follows:

$$Sim(U_a, U_b) = \frac{\overrightarrow{U_a} \cdot \overrightarrow{U_b}}{||\overrightarrow{U_a}|| \times ||\overrightarrow{U_b}||} \quad (4.9)$$

$$Sim(U_a, U_b) = \frac{\sum_{p \in P} U_a \cdot U_b}{\sqrt{\sum_{p \in P} (U_a)^2} \sqrt{\sum_{p \in P} (U_b)^2}} \quad (4.10)$$

Where:

$Sim(U_a, U_b)$  = cosine similarity of two vectors

P = the set of courses that have been rated by user  $U_a$  and  $U_b$

The algorithm firstly calculates the dot product that is the sum of the products of the two vectors. However, as the dot product is sensitive to the magnitude, it might show that two vectors with a similar direction are dissimilar to each other, owing to one having a larger

magnitude than the other. In view of this, the value needs to be normalised by dividing the product of the lengths of the two vectors together and calculating the cosine similarity by using the unit vector rather than the normal vector. To present this, table 4.3 is proposed as an example of the user's recommendation histories for  $U_a$ ,  $U_b$ .

User( $U_a$ )		User ( $U_b$ )	
Course name	Course rate	Course name	Course rate
C1002	4	C1007	5
C1004	2	C1005	2
C1003	5	C1004	5
C1001	4	C1001	4
C1005	3	C1006	4

Table 4.3 Example of  $U_a$ ,  $U_b$  recommendations history

Eq.4.10 was used to calculate the similarity between  $U_a$ ,  $U_b$  recommendations histories as the following:

$$\text{Sim}(U_a, U_b) = \frac{\sum_{p \in P} U_a \cdot U_b}{\sqrt{\sum_{p \in P} (U_a)^2} \sqrt{\sum_{p \in P} (U_b)^2}}$$

$$\sum_{p \in P} U_a \cdot U_b = (4 * 0) + (2 * 5) + (5 * 0) + (4 * 4) + (3 * 2) + (0 * 5) + (0 * 2) = 32$$

$$\sqrt{\sum_{p \in P} (U_a)^2} = \sqrt{(4 * 4) + (2 * 2) + (5 * 5) + (4 * 4) + (3 * 3)} = \sqrt{70} = 8.36$$

$$\sqrt{\sum_{p \in P} (U_b)^2} = \sqrt{(5 * 5) + (2 * 2) + (5 * 5) + (4 * 4) + (4 * 4)} = \sqrt{86} = 9.27$$

$$\text{Sim}(a, b) = \frac{32}{8.36 \times 9.27} = 0.412 \times 0.5 = 0.206$$

In order to obtain the final similarity value between  $U_a$ ,  $U_b$ , Eq.4.7 has been used as the following:

$$US(U_a, U_b) = \text{ontology similarity} + \text{recommendation history similarity}$$

$$US(U_a, U_b) = 0.35 + 0.206$$

$$US(U_a, U_b) = 0.556$$

#### 4.4.2.2 Ontology based K-nearest neighbour algorithm

The k-nearest neighbour users of the active user (target user) must be determined in order to make a recommendations list by CF. To achieve this result, a new algorithm is proposed, OKNN algorithm that combines the ontology similarity of the user profile attribute and the item rate when the recommendation history is applied. The k-nearest neighbour users to the target user are found by searching only those who exist among the same group, rather than all the users. For instance, if the target user has a main subject of Computer Sciences and their major is Computer Programming, the nearest neighbour will search for all the users who have Computer Sciences as a main subject in their profiles. In addition, not all of the groups are searched in the User-Clustering attribute of the items selected. The user similarity, based on Eq. (4.11), has been used to locate who is the neighbouring user to the target user. To find the top k-nearest neighbour to the target user, the users' similarity score needs to be ranked. A common rate problem that was faced for the top k-nearest neighbour was that the same item had been rated by different values respectively. In order to solve this problem, the following formula has been proposed:

$$\text{Average weight score} = \left( \left( \frac{\text{ARW C} \times (\text{KNNW} - \text{Omax} \times K)}{\text{KNNW}} \right) + \text{Oc} \times K \right) / 100 \quad (4.11)$$

Where:

KNNW = KNN weight in the final scoring function

ARW c = average weight of the rate for the current course \*100%

Omax = the maximum occurrence of the rate in the recommendation history of all the top N users

K = constant (e.g. 2)

Oc = the number of occurrences of the current course has been rated

Table 4.4 describes a real sample of the system experiment that shows the top k-nearest neighbour courses and their rates and how the common rate problem can be solved by using Eq.4.11. There are five users who have received recommended courses and rated them by a value range of between 1 and 5. The threshold value of the recommendation rate has been set at the rate of  $\geq 3$ , so any courses rated lower than that threshold value will be removed from the list as shown in table 4.5.

U1courses	RateU1	U2cousese	RateU2	U3course	rateU3	U4course	rateU4	U5course	rateU5
C001	3	C003	3	C005	4	C003	5	C008	5
C005	4	C030	3	C003	4	C021	2	C005	4
C010	3	C111	4	C044	3	C049	3	C111	3
C020	5	C005	4	C020	2	C111	4	C010	4
C022	2	C033	5	C050	5	C122	2	C011	2

Table 4. 3 Table 4.5 sample of top k-nearest neighbour courses and their rates

U1course	RateU1	U2courseses	RateU2	U3courses	rateU3	U4courses	rateU4	U5course	rateU5
C001	3	C003	3	C005	4	C003	5	C008	5
C005	4	C030	3	C003	4	C021	3	C005	4
C010	3	C111	4	C044	3	C049	3	C111	3
C020	5	C005	4			C111	4	C010	4
		C033	5	C050	5				

Table 4. 4 Table 4.5 removed courses and their rate if rate  $\geq 3$

In the scenario above, there are 21 courses which needed to be listed as KNN course recommendations. However, the following issues arose:

1. The courses have been rated many times with the same rate value C005 appearing four times. It needs to feature in the list only once but how many times it has been rated also needs to be considered.
2. The course has been rated many times with different rate values as C003 and C111 which has been rated by 3, 4, 5 and 4, 4, 3 respectively.

The Eq.4.11 has been used to solve the above issues. In the case of C005, the variable values in the Eq.4.11 will be as the following:

$$ARW_C = (0.2+0.2+0.2+0.2)/4 = 0.2 \times 100\% = 20$$

Omax = 4 (based on table 4.4 it can be seen that the maximum of occurrence of rate in all the Top N users' is 4)

K=2 (we can use any value of constant here we used 2)

Oc= 4

$$\text{Average weight score} = \left( \left( \frac{ARW_C \times (KNNW - O_{max} \times K)}{KNNW} \right) + Oc \times K \right) / 100$$

$$\text{Average weight score} = \left( \left( \frac{20 \times (30 - 4 \times 2)}{30} \right) + 4 \times 2 \right) / 100$$

$$\text{Average weight score} = 0.226$$

Moreover, in case C003 the variables values in the Eq.4.11 will be as the following:

$$ARW_c = (0.1+0.2+0.3)/3 = 0.2 \times 100\% = 20$$

Omax = 4 (base on KNN table it can be seen that the maximum of occurrence of rate in all the Top N users' is 4)

$$K=2$$

$$Oc= 3$$

$$\text{Average weight score} = \left( \left( \frac{ARW_c \times (KNNW - O_{max} \times K)}{KNNW} \right) + Oc \times K \right) / 100$$

$$\text{Average weight score} = \left( \left( \frac{20 \times (30 - 4 \times 2)}{30} \right) + 3 \times 2 \right) / 100$$

$$\text{Average weight score} = 0.206$$

The proposed method improves the scalability and accuracy, leading to an improvement in the performance of the algorithm. In this algorithm, the similarity between ontologies is used to compare the target user profile to other users to obtain k-NN users. In this method of similarity, the conceptual similarities are considered when measuring the similarity between two ontologies. The conceptual comparison level includes the comparison between two taxonomies and the comparison of relations between the corresponding concepts of the two taxonomies. After producing the k-nearest neighbour users, all courses that have been selected by the neighbour users, but have not been selected by the target user, are recommended to the target user. The final step in the method is that the final recommendation list can be presented to the active user according to a hybrid recommendation list from both the CBF and CF filters based on a weighted approach. The steps of this algorithm are presented as follows in OKNN algorithm.

---

	(OKNN) ALGORITHM
1:	For user Uc Get user profile and create vector
2:	<b>while</b> there are Users to compare U <b>do</b>
3:	Create vector for U
4:	Calculate the Similarity between U and Uc by using Formula (4.7)
5:	Sort the nearest neighbour list
6:	Get the top 5 nearest neighbour
7:	<b>for</b> each user in the top 5 list <b>do</b>
8:	<b>for</b> each course in the user's recommendation history
9:	<b>If</b> C rate $\geq 3$ <b>then</b>
10:	add the C to the KNN list
11:	<b>end if</b>
12:	<b>end for</b>
13:	<b>end for</b>
14:	<b>for</b> each C in the KNN list <b>do</b>
15:	Calculate the C rate using Equation (4.11)
16:	Update the KNN list with the new score
17:	<b>end for</b>
18:	<b>end while</b>

---

#### 4.4.2.3 Final Scoring Algorithm

The proposed approach to filtering combines CBF and CF with ontology to recommend courses to the user. For the new user, the system will recommend courses based on his/her profile. The recommendation process will begin based on the OPCR algorithm by creating a vector of users and courses.

The final recommendation list is produced by using the final scoring function (FSF). FSF combines the similarity score of a content-based filtering list and a collaborative filtering list. Moreover, other factors will be added to the final score as well, such as the university ranking and the NSS score as shown in the Eq. (4.12). The value of the final score function similarity should be between the range (0-1). The weight percentage for each part in the FCF (CBF, CF, university rank, NSS score) is 50%, 30%, 10%, 10%, respectively.

$$\text{Final Scoring Function (FSF)} = (\text{CBF} \times (50\%)) + (\text{CF} \times (30\%)) + (\text{University rank} \times (10\%)) + (\text{NSS score} \times (10\%)) \quad (4.12)$$

OPCR ALGORITHM	
1:	Calculate course score based on the user's profiles and query term
2:	Calculate course NSS score similarity using formula (6)
3:	Calculate university rank similarity by using formula (5)
4:	Calculate course fees similarity by using formula (4)
5:	Rank and get the top 10 courses which have the highest similarity score
6:	Recommend the user by the top 10 courses
7:	<b>If</b> the user chooses any of the recommended courses <b>then</b>
8:	Use OKNN algorithm to determine the five nearest neighbours of the current user $U_c$ calculate course scores
9:	add top 10 courses to the recommendation list
10:	<b>end if</b>
11:	Return the refined recommendations to the user

## 4.5 Summary

This chapter has discussed the recommendation algorithms and ontology model in OPCRa which is designed to address the problem of information overloading in the education domain. A novel hybrid solution was proposed, which combines content-based filtering method and collaborative based filtering method with ontology. Each of these methods has different scores calculated at each stage.

Using a hybrid filtering method can help to tackle the drawbacks of both the CBF and CF. On the other hand, utilising ontology in the recommendation system allows the OPCRa to provide more accurate recommendations and to increase user satisfaction. In the next chapter, more details are provided regarding the efficient performance of the OPCRa when compared with other current approaches.

# CHAPTER 5 IMPLEMENTATION AND RESULTS

*“I have not failed. I've just found 10,000 ways that won't work.”*

Thomas A. Edison

To evaluate the OPCR framework, a Web based prototype system has been implemented using Java programming language. All modules in the OPCR framework have been implemented using open source tools in the back end. The prototype system was stamped as OPRCourse which is a java based application with a web interface. The ontology model has been constructed and validated by using a protégé tool. All the data that has been used in the experiment is real time information extracted from multiple sources. The prototype application was publicly accessible through the University of Portsmouth subdomain at oprcourse.ee.port.ac.uk. The site remained live until the analysis was completed. For data integration in the course domain and job domain, a web crawler based ontology was implemented. The crawler can be used in many different domains with certain changes to the back end code in order to match the domain's requirements. All codes are available for developers and researchers on the author account in github.com under the link: <https://github.com/mohammediraq/OPCR>.

This chapter introduces the implementation of the prototype system, which begins by presenting the full operational functions, starting with the data collected from educational institutions such as UCAS, continues with the construction of the ontology model and mapping ontologies and finishes with scoring and recommendations.

The prototype application was implemented and run on an Intel(R) Core(TM)2 Dup CPU processor, with a CPU of 3.20 GHz and 16 GB of RAM, under Windows 7. HTML was used for the system interface, and a MySQL server was used to allocate a system dataset and user ratings. A protégé tool was also used to evaluate the ontologies built in the system

## 5.1 Data Source and Configuration

In the system, all the data that has been used to implement OPCRa was from free open sources resources. Course information was extracted from UCAS.com website, and job information was extracted from Indeed.com website. NNS score information was collected from officeforstudents.org.uk website, and university ranking information was extracted from guardain.com. However, most of this information was unstructured. In order to organise this, a web crawler was built and customised. The collected data was used to construct the item

ontology (courses and jobs), based on acquired knowledge. There was no existing specific dataset which could be applied to implement in the design model. A dataset called ontologyset was therefore created which included courses extracted from UCAS.com. However, there was no need for an established benchmark dataset to evaluate the performance of the OPCR. The system metadata included close to 21,000 online courses in ontologyset, covering 70 diverse subject areas that had been archived from UCAS.com. These were focus chosen and downloaded from different departments at various universities and colleges in the United Kingdom for testing purposes. The breakdown was to select 20 of these subject areas with a number of courses, but it was decided to use the computer sciences and business management courses. The courses subjects were classified based on HECoS (The Higher Education Classification of Subjects). Please refer to Appendix A and tables A.1 and A.2 which show the classification of the subjects in the fields of information technology and business and administration. The courses in the dataset cover all postgraduate academic levels which yielded a representative set that included a wide range of courses offered at different universities. For job information, the Indeed.com website was used as a source in order to extract job information. This information included the job title, description, salary, location and user reviews. For test purposes, any jobs related to CS and BAM courses were extracted.

### 5.1.1 Data Collection Module

Web server environments place particular requirements on the software that they integrate with. Typical Java web application servers, such as Tomcat, handle each HTTP request in a separate thread. When a request comes in, the request handler is activated; if it needs database access, it opens a connection (typically from a connection pool), carries out the required processing and returns the database connection to the pool. Some architectures lease database connections to request handlers on an even shorter-term basis, such as once for each database operation.

### 5.1.2 Web Crawler

The crawler was developed using a Java framework (version JDK1.8) as well as NetBeans (version 8.02) being utilised as a development tool on a Windows platform as shown in Appendix C. The system runs on any standard machine and does not require any specific hardware in order to run effectively. Approximately 6000 web pages were downloaded from UCAS, and their links were recorded in the database. This experiment focuses on a crawling topic of a “computer sciences course”. The reference ontology model was created using a

protégé tool. A web-based user interface was used to prompt the user to give a query as shown in Fig.5.1.

Figure 5.1 Web crawler interface

In order to measure the performance of the proposed approach, a harvest rate metric was used to measure the crawler performance, and a recall metric was used to evaluate the performance of the webpage classifier.

The harvest rate, according to (Gautam Pant & Padmini Srinivasan, 2006; Pant & Menczer, 2003), is defined as the fraction of the web pages crawled that are relevant to the given topic. This measures how well irrelevant web pages are rejected. The formula is given by:

$$\text{Harvest rate} = \frac{\sum_{i \in V} r_i}{|V|} \quad (5.1)$$

Where  $V$  is the number of web pages crawled by the focused crawler in current;  $r_i$  is the relevance between the web page  $i$  and the given topic, and the value of  $r_i$  can only be 0 or 1. If relevant, then  $r_i = 1$ ; otherwise  $r_i = 0$ . In the Fig. 5.3 the performance comparison between the proposed crawler which is based ontology and a traditional crawler is shown.

In Fig.5.2, the  $x$ -axis denotes the number of crawled web pages and the  $y$ -axis denotes the average harvest rates when the number of crawled pages is  $N$ . According to Fig. 5.3, the number of crawled web pages increases and it can be seen that the number of crawled web pages of WCO is higher than that of a simple crawler which has not used ontology in the crawling process. Moreover, the harvest rate of a simple crawler is 0.42 at the point that corresponds to 100 crawled web page in Fig. 5.2. However, the values indicate that the harvest rate of the

WCO is 0.6 and 1.3 times larger than that of the simple crawler. The recall metric is the fraction of relevant pages crawled and used to measure how well it is performing in finding all the relevant webpages (Pant et al., 2004).

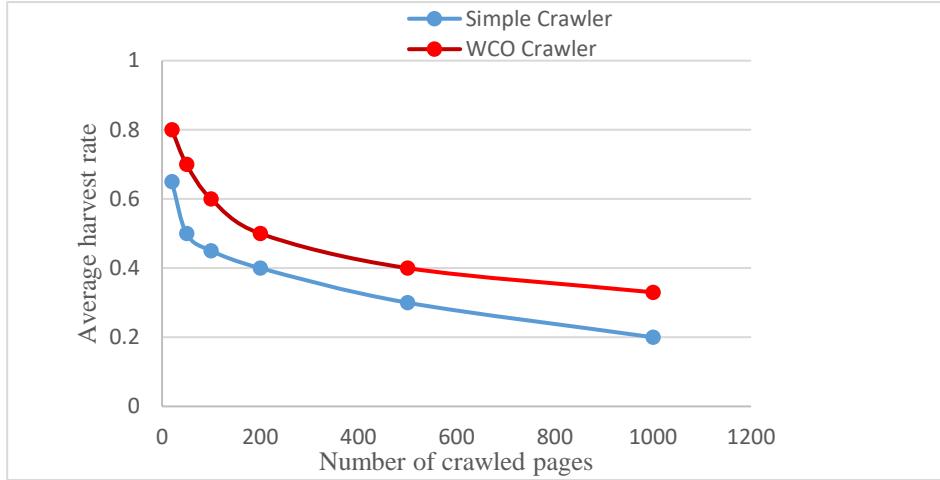


Figure 5.2 Performance comparison between ontology crawler and traditional crawler.

The recall metric is the fraction of relevant pages crawled and used to measure how well it is performing in finding all the relevant web pages. To further understand how the recall metric can be applied, let  $RS$  be the relevant set in the web and  $S_t$  be the set of the first pages crawled. The formula of the recall metric will be as follows:

$$\text{Recall metrics} = \frac{|RS \cap S_t|}{|RS|} \quad (5.2)$$

Fig. 5.3 presents a performance comparison of the average recall metrics for a simple crawler and the proposed crawler for five different topics. The  $x$ -axis in Fig. 5.3 denotes the number of crawled web pages and the  $y$ -axis denotes the average recall metrics when the number of crawled pages is  $N$ . It was realised that the average recall of the WCO is higher than that of the simple crawler.

Based on the performance evaluation metrics, the harvest rate and the recall metrics, it can be concluded that the WCO has a higher performance level than that of a simple crawler.

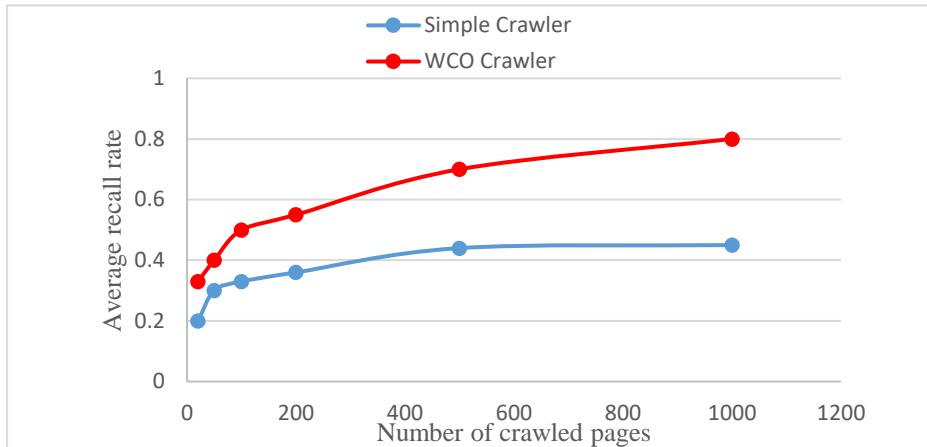


Figure 5.3 Performance comparison between ontology crawler and traditional crawler

## 5.2 Application of OPCR

The OPCRa application was implemented using Java programming language on the server-side and web-based for the client aspect as shown Fig. 5.5. To implement the use of OPCRa, a proof of concept system was developed for master's courses in computer sciences and business and management. The back-end OPCR was a Java-based application with a web interface. All the information was read from and sorted to a MySQL database.

### 5.2.1 OPCR Interface

In the client part of the OPCRa, the user interacts with the system via the Graph User Interface (GUI) which was implemented using Java as a backend and web-based in the front end as shown in appendix C Fig C.1 and the interface flow diagram shown in Fig 5.4. Interface flow diagrams show the high levels relation and interaction between the interface objects of an application, and these diagrams help to validate the design of the user interface and can be used to determine if the user interface has been designed consistently. The interface flow diagrams in Fig 5.4 models the flow between the pages of the OPRCourse system. It describes exactly how the user will navigate through the user interface for each possible user input event. Some of the edges in the figure have a word pair associated with it. The GUI sends the data to/receives data from the protocol handler and also sends/receives data according to the user's actions. The protocol handler is responsible for setting up communication facilities and to send/ receive commands formatted in a way that the server can interpret.

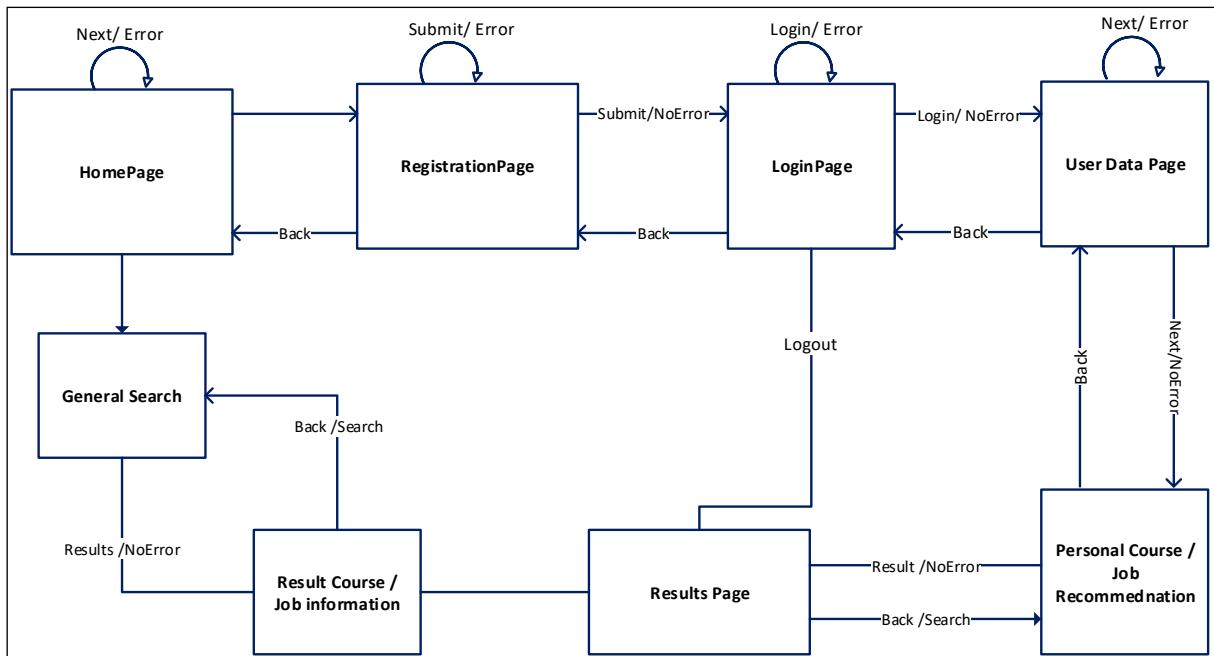


Figure 5.4 Interface data flow diagram

OPCR users can be students or developers. In the case of a user, the interface will be used for the following activities:

- General course search**

OPCR used a keyword-based similarity technique in the case where the user wants to search for a specific course name in the database. The HTTP request will handle the user query and send it to the search function in the core component in the back end. In order to calculate the

#	Course Name	Field of Study	Course main subject	University Name	Course Fee	Course Duration	Course Location	Course URL
1	Information Technology	Information technology/	Computer graphics	University of Aberdeen King's College Aberdeen AB24 3FX	6,300	12 months	- AB24 3FX	<a href="#">course website</a>
2	Artificial Intelligence	Information technology/	Computer graphics	University of Aberdeen Student Recruitment and Admissions King's College Aberdeen AB24 3FX	6,300	12 months	King's Campus - AB24 3FX	<a href="#">course website</a>
3	Arts, Media and Computer Games	Information technology/	Computer graphics	Abertay University Bell Street Dundee DD1 1HG	4,195	14 months	Abertay Campus - DD1 1HG	<a href="#">course website</a>

Figure 5.5 Snapshot of the general course search page

similarity between the user query keywords and the courses tiles in the database, TF-IDF has

been used. After calculating the similarity, the user will receive a list of course suggestions as a result of his/her request as shown in Fig 5.5 which represents a snapshot of the general course search page

- **Create a user account**

In order to provide the user with personalised recommendations, the user needs to provide the system with certain basic information in order to generate a user profile for him/her. User information included two types of information; personal and academic background. This information was used in the CBF method to create a user vector by which to measure the similarity between the concepts in the user profile and the course profile. Likewise, the system will work to match user profile features with job information from the database and map this with course features. To map course features with a student profile and a job profile, the ontology mapping approach as defined in section 4.2.5 was used. Since the user completes the registration page, the information in the user profile will all be automatically mapped with the ontology model. For example, for personal information such as an address, if the user types in the city name as “Portsmouth” in the postal address field, the system will link this city automatically with the region to which it belongs to, which is “Hampshire”. Furthermore, the user profile will map with all the courses at universities throughout this region. For academic information, the system will present all the course subject information to the user in order to select the field of study, the main subject of the course and the major subject of the course. In the case where the user did not indicate the course subject in the presented options, OPCR used a dynamic ontology construction approach which allows users to add their subject ontology, and this will be linked to the ontology model in the framework. Fig. C.2 in the appendix C show the snapshot of the registration page.

- **Obtain ontology-based recommendation list**

After creating a user profile which includes all the user features, the user needs to carry out certain further actions in order to obtain a list of recommendations. Users have different levels of financial budgets and thus are looking to enrol on the university course that meets with their own budget. OPCRa includes a feature that allows users to customise the search by adding a maximum course fee which matches their financial budget. As mentioned in section 4.3.1, course fees were one of the features that were used to calculate the similarity in the CBF method. The recommendation results will be affected by changing the value of the course fee. The CBF algorithm used formula 4.4 to calculate the similarity between a user maximum fee

and course fees. The user also needs to fill the value of NNS score which is expected of the educational institution that provides the course. The final step that needs to be taken in order to receive a list of recommendations is to click the search button on the ontology search page, as shown in Fig 5.6. OPCR provides the users with five course recommendations which are the

Most relevant courses. Users are asked to provide a rating for each course in the recommendation list in order that this can be used in the CF algorithms and thus recalculate the

The screenshot shows a web browser window with the URL [localhost:8080/Ontology/search.html](http://localhost:8080/Ontology/search.html). The page is titled "Welcome MOHIB00012". It features a search bar with placeholder text "Search for Artificial Intelligence" and a note "Enter the course name / search keyword". Below it are fields for "Acceptable Fees" (set to 20000) and "University NSS score" (set to 80). A "SEARCH" button is present. The main content area displays two sections: "We got courses recommended for you" and "We got jobs recommended for your course major". The "Courses" section lists two entries:

Course ID	Course Name	Field of Study	Course Major subject	University Name	Course Fee	Course Duration	Course Location	Course URL	Rate
2054	Artificial Intelligence	Information technology/	Artificial Intelligence	University of Edinburgh	4,370	3 years	Central course campus - EH8 9J5	course website	<input type="button" value="▼"/>
1954	Data Science and Computational Intelligence	Information technology/	Artificial Intelligence	Coventry University	7,750	1 year	Main Site - CV1 5FB	course website	<input type="button" value="▼"/>

Each row has a "Course Details" button. The "Jobs" section lists two entries:

Job Details	Job ID	Job Title	Company	Salary	Reviews
<input type="button" value="Job Details"/>	J18126	Software Developer (Artificial Intelligence)	C4Commerce London	£35,000 - £45,000 a year	50 reviews
<input type="button" value="Job Details"/>	J18127	Artificial Intelligence Engineer	Reverie Digital Recruitment	£65,000 - £75,000 a year	44 reviews

Figure 5.6 Snapshot of ontology-based course search page in OPCR

course filtering to provide the user with more accurate recommendations taken into account the similar users' opinions. To do this, OPCR used a new algorithm called (OKKN) which was described in section 4.3.22. The features of each recommendation will provide a comprehensive knowledge regarding each course because they will include the following information (course name, field of study, major subject of the course, university name, course fee, course duration, university location, course URL for obtaining more details).

In the case of the developer or administrator, the user will be able to perform many activities through the use of the administrator page. OPRCourse allows users, through the administrator page, to manage all the features in the framework. The user needs to work at the front end, and all requests will be sent to the core components at the backend. Feature management included the features that are related to the ontology model, item feature weights, user feature weights and algorithms weights in the final furcation scoring. Fig. 5.7 shows a snapshot of the

administrator page and Appendix B provides more detail of each activity of the administrator page.

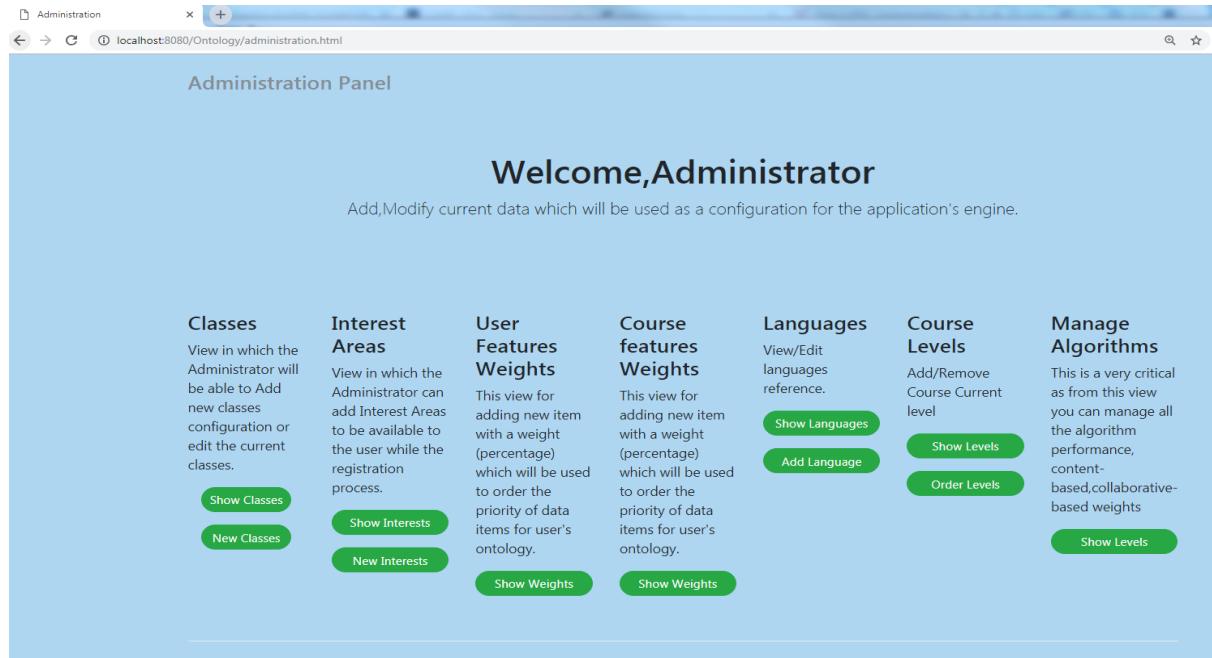


Figure 5.7 Snapshot of administrator page in OPRCourse

### 5.2.2 Ontology model

This section features the tools and technologies used for the implementation of the various modules of the ontology model that were displayed in chapter 4. The modules were implemented by using a collection of JAVA libraries. In order to facilitate the development, all the outputs of the methods were stored in the database, so that it was possible to operate in the middle of two steps, such as ontology construction and ontology mapping. The development was carried out using the open source development platform, NetBeans, in combination with various open sources tools such as a protégé tool for ontology editing and a Jena framework that allows the system to work with OWL file from Java classes in a transparent way.

As mentioned in section 3.3.1, the information was extracted from multiple sources and then structured in the database. In order to build an ontology from the information in the database tables, the concepts for each ontology domain were extracted. In order to be able to work with persistent ontology models to form a program, it was necessary to use a certain frame to act as a gateway between the application and the relational database. Jena is one of the open source java frameworks used for building semantic web applications.

In this implementation, two different approaches to working with ontology models were used. Firstly, the domain ontologies, such as the course domain or job domain, were loaded directly from the ontology file as memory models because this is the most productive way of working with non-persistent models. Secondly, user profiles were loaded utilising persistent ontology models because, in this case, data needs to be managed and modified the data.

#### 5.2.2.1 Ontology Construction Module

Jena API is a Java API (Application Programming Interface) framework (Jena 2011) that provides classes and interfaces by to construct ontologies using the set of extracted semantic concepts and their corresponding relationships. The constructed ontology is represented in the form of a semantic mark-up language called Web Ontology Language (OWL). Protégé 5.2 is the ontology visualisation and editing tool with the vizOWL plug-in which provides the platform to visualise the automatically constructed topic ontology successfully and also generates the concept class hierarchy with the final hierarchy of classes and subclasses depicted.

With regard to the limitations mentioned in section 3.3.4.1, the ontology model should be able to create the ontology applying OWL syntax. Therefore, OWL models from the Protégé API were used to build the created ontology. OWL allows the ability to create, query or delete components of OWL ontologies such as classes, properties or individuals. For the purpose of this project, an OWL model was used for storing the structure of the generated ontology and subsequently, the content of this model was written into an OWL file. The system database included structure tables for the course ontology, user ontology and job ontology. In order to validate domain ontology model two fields of study were selected which were computer sciences subjects and business and administrative studies. In the case of computer sciences subjects, based on the standard classification made by HECoS in UK universities, it was noticed that seven main subjects of university courses were classified from the root subject computer sciences such as (information system, computer science, artificial intelligence, health information, computer generated visual and audio effects, software engineering and games) as shown in Fig. 5.8.

The main subjects are represented in the ontology as subclasses of the main class. Each of the

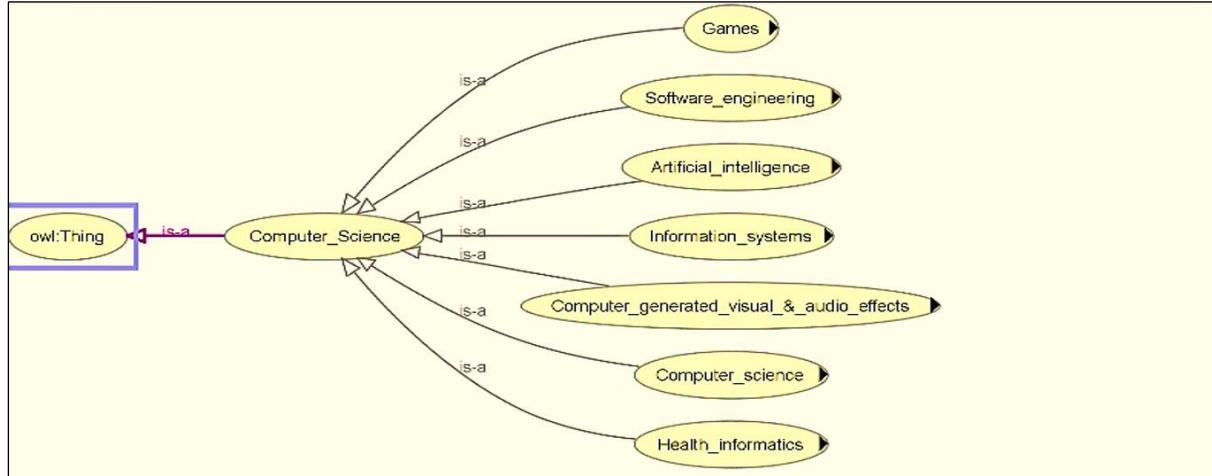


Figure 5.8 OWL Viz view of Computer Science Classes generated by Protégé

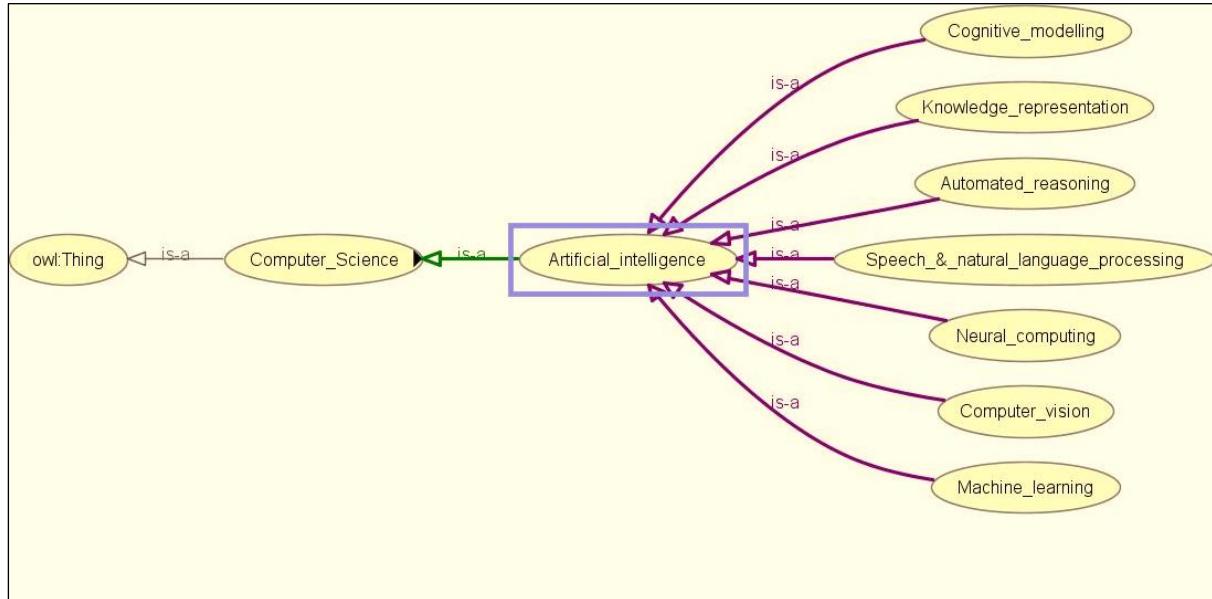


Figure 5.9 OWL Viz view of Artificial Intelligent Classes generated by Protégé

subclasses has many major subjects which are important to know about when a student wants to make a course selection because each of the major subjects may lead to a different type of career. For instance, artificial intelligence as a main subject in the field of computer sciences contains many major subjects such as machine learning, cognitive modelling, neural computing, knowledge representation, automated reasoning, speech and natural language processing, computer vision, as shown in Fig.5.9.

Fig. 5.10 shows a sample of concepts hierarchy of the classes and subclasses for computer sciences subjects.

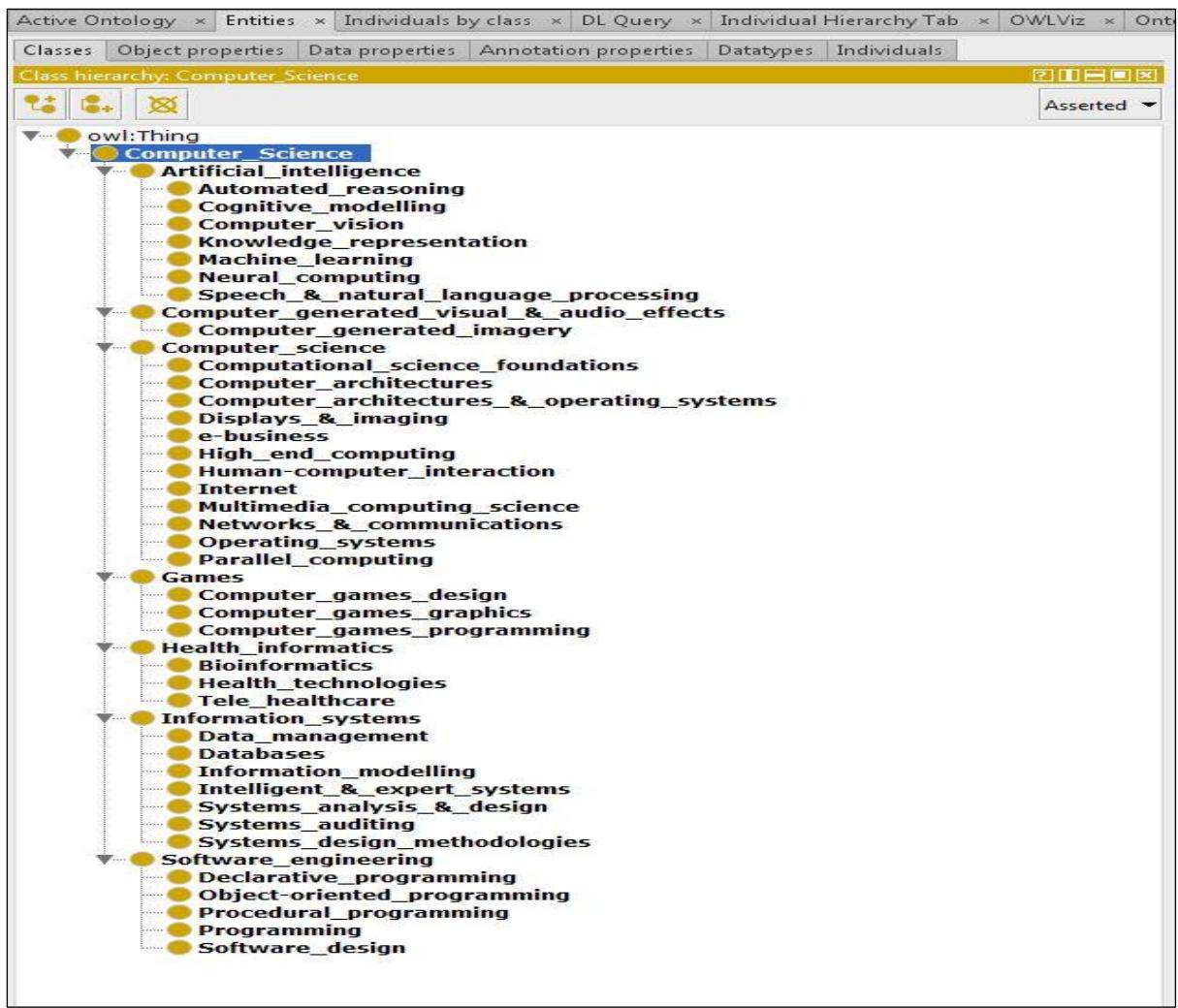


Figure 5. 10 Computer science subjects hierarchy

### 5.2.2.2 Ontology Mapping Implementation

This section has discussed the implementation process of ontology mapping in the framework. As mentioned in chapter 3, OPCR has three main ontology domains which are course ontology, student ontology and job ontology. Each of these ontologies has a number of classes, subclasses and properties. To implement mapping ontology, Java code has been used with supporting Jena API to read OWL file. To validate the mapping approach Protégé tool used as ontology editor, each relevant classes of course ontology have been mapped to student ontology and job ontology by subject properties. For instance, class `city` course ontology has been mapped to all `university_location` class in course ontology, `postalAddress` class in student ontology and `JobLocation` class in the job ontology. Moreover, also, the class `MajorSubject` in course ontology has been mapped to a `course_information` class in the course ontology,

*EducationalInformation* in student ontology and *AcademicRequirements* in the job ontology as shown in Fig. 5.11.

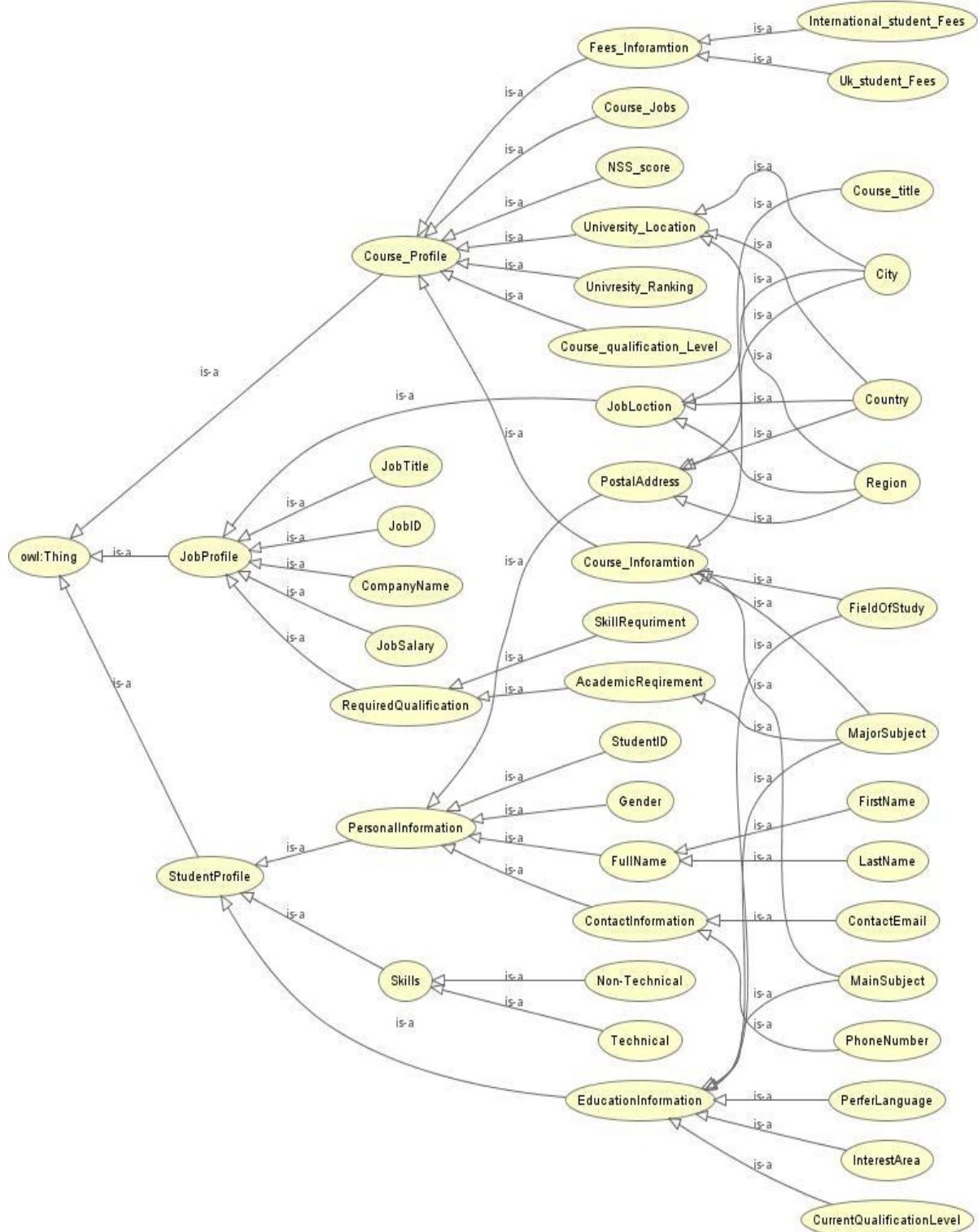


Figure 5.11 Ontology mapping implementation

### 5.2.3 Recommendation Producing

The main aim of OPCR is to reduce information overloading of the suggestions that return when a user conducts a search of university courses. It also provides users with personalised recommendations which meet with their individual needs. To implement the performance of the proposed approach in this thesis, a database was used that has two categories of the field of study subject with 70 main courses subjects from 136 universities in the UK. OPCR is flexible which means it can be applied in a different domain, but a master's science course was used in this implementation as the qualification level sample. As per the implementation process of the recommendations algorithm that was described in chapter 4, the content base Filtering was adopted as a primary approach because the new user has as yet no user history. The final course recommendation list combines the results of content based filtering and collaborative based filtering and also takes into account the ontology mapping between the course profile, the

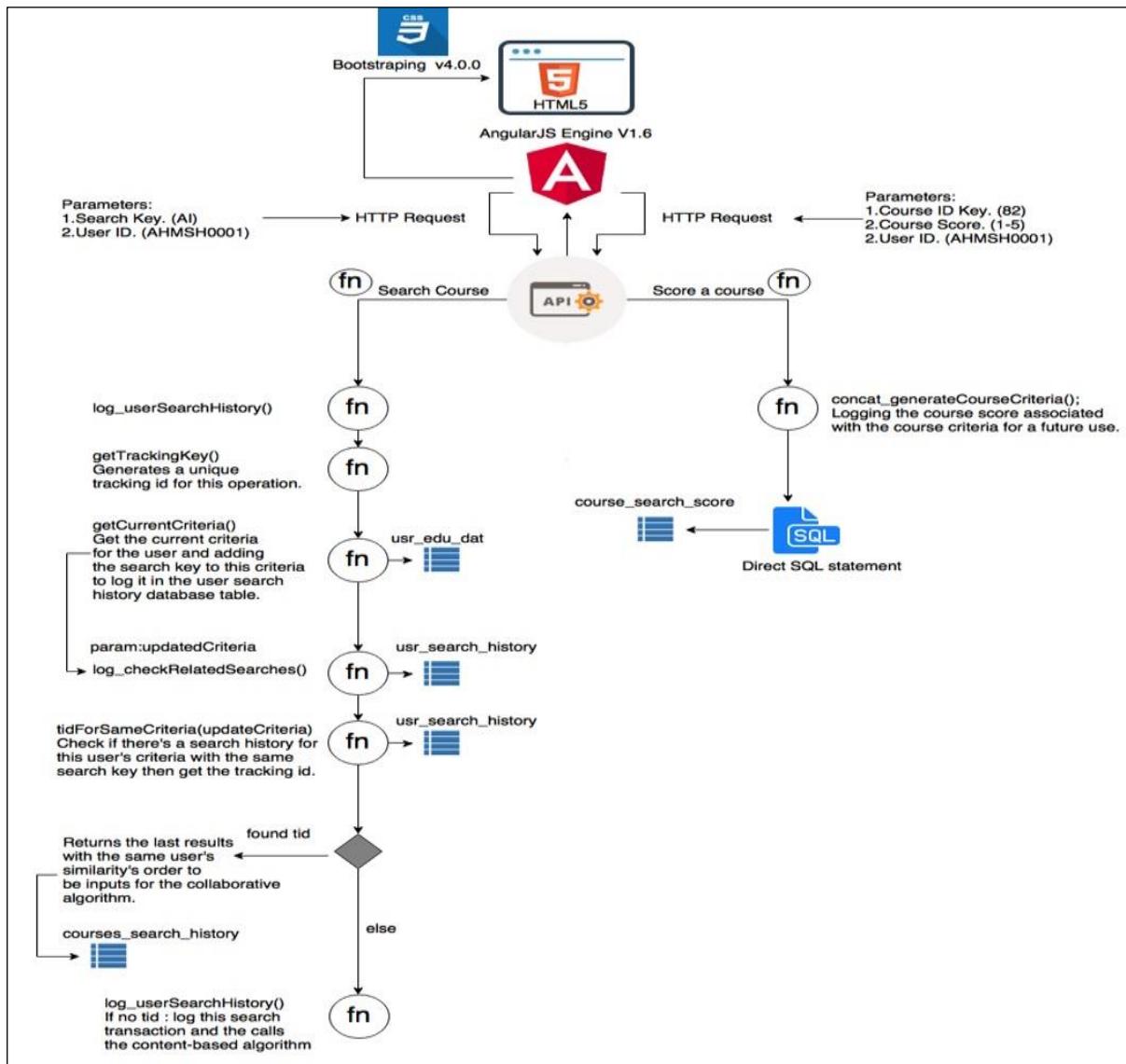


Figure 5. 12a Recommender engine functions flow diagram

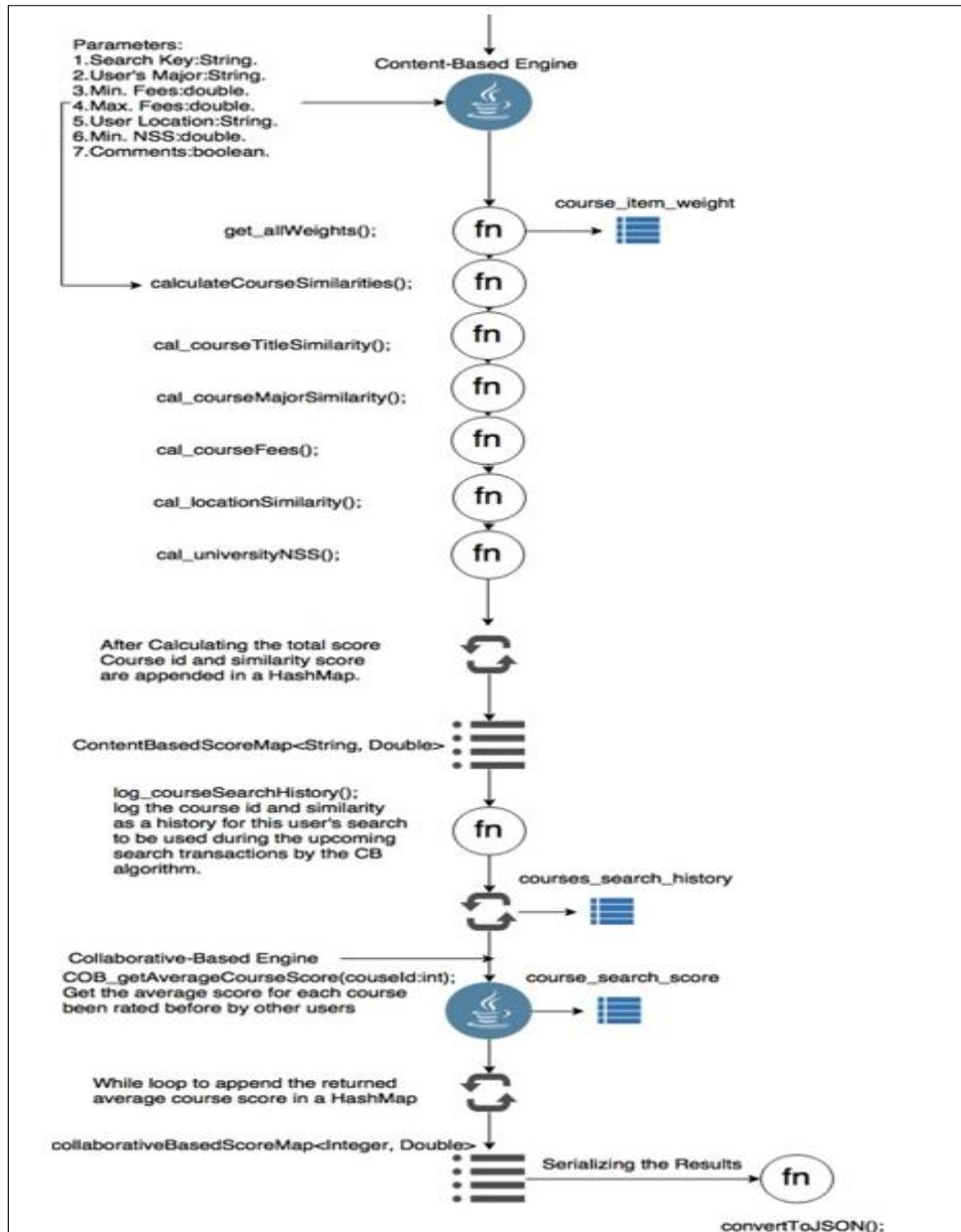


Figure 5. 12b Recommender engine functions flow diagram student profile and the related career prospects of the course's major subject in order improve the quality of the recommendations and to increase user satisfaction. Fig (5.12a, b) shown the CBF and CF functions implementation process flow diagram. The following subsections discuss the implementation process in each method.

### 5.2.3.1 Content Based Filtering

This section explains the content-based recommendation component of the recommendation system as defined in the content based filtering section in chapter 4. The implementation process begins after users have completed the information required on the registration page and clicked the *submit* bottom. The user information is stored in a MySQL database.

Content based filtering works to generate a user (student) profile vector and an item (course) profile vector. Each vector has a number of features and, based on experiments, weights for each feature adjusted and assigned in both the user and item vectors. In Appendix B, an implementation process diagram for main feature similarity calculation in CBF can be found. Furthermore, to recommend an item in CBF, the similarity between the user profile and all the courses profile in the database was computed. To recommend the most relevant course, the similarity degree was ranked from high to low.

The disadvantage of a traditional CBF occurs when, if there is a large number of items in the database, the similarity calculation time is very high, and the process uses more space in the process memory. OPCRa used an ontology technique that overcame this problem by computing the similarity of the levels for the concepts hierarchy of the user vector and the course vector. For instance, if the user vector features include the following information - computer sciences as the field of study, artificial intelligence as the main course subject and machine learning as a major subject of the course - CBF filtering will calculate the similarity with courses in the database only if those courses have the same field of study because the courses have been structured in the database according to the ontology model. For similarity calculation in CBF, cosine similarity (Eq. 4.3) and feature matching have been used. A user using the system for the first time will receive an initial recommendation list, based on CBF, of the five top recommendations required to be rated by him. The rate scale in this thesis used numbers from

5 to 1, with 5 indicating strongly recommended and 1 as a weak recommendation as shown in Fig 5.13.

Course ID	Course Name	Field of Study	Course Major subject	University Name	Course Fee	Course Duration	Course Location	Course URL	Rate
2054	Artificial Intelligence	Information technology/	Artificial intelligence	University of Edinburgh	4,370	3 years	Central area campus - EH8 9JS	course website	<div style="border: 1px solid #ccc; padding: 2px;"> <input type="button" value="▼"/> </div> <div style="margin-top: 5px;"> <input checked="" type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> </div>
1954	Data Science and Computational Intelligence	Information technology/	Artificial intelligence	Coventry University	7,750	1 year	Main Site - CV1 5FB	course website	<div style="border: 1px solid #ccc; padding: 2px;"> <input type="button" value="▼"/> </div> <div style="margin-top: 5px;"> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> </div>

Figure 5.13 Snapshot of recommended courses rating

All the rates will be saved in a separate table in the database named *course\_search\_score* and linked to the user profile by the user ID as shown in Fig 5.14. Collaborative based filtering will use these rates to find the top user nearest neighbour to the target user. In the following subsection, the collaborative based filtering process will be discussed in more detail.

recid	course_id	course_score	course_criteria	user_id	transaction_id	time_stamp
895	1924	5	1924-Information technology-Computer science-Artificial intelligence	MOHIB00012	NULL	2018-04-09 15:47:41
896	2014	4	2014-Information technology-Computer science-Artificial intelligence	MOHIB00012	NULL	2018-04-09 16:03:39
897	2006	1	2006-Information technology-Computer science-Artificial intelligence	MOHIB00012	NULL	2018-04-09 16:04:02
898	1868	4	1868-Information technology-Computer science-Artificial intelligence	MOHIB00012	NULL	2018-04-16 15:03:34
899	1904	5	1904-Information technology-Computer science-Artificial intelligence	MOHIB00012	NULL	2018-04-16 15:03:37
900	2006	5	2006-Information technology-Computer science-Artificial intelligence	MOHIB00012	NULL	2018-04-16 15:03:40
901	3961	4	3961-Business and management-Business studies-Accounting	SARM100014	NULL	2018-04-20 21:04:16
902	4417	4	4417-Business and management-Business studies-Accounting	SARM100014	NULL	2018-04-20 21:04:19
903	4151	3	4151-Business and management-Business studies-Accounting	SARM100014	NULL	2018-04-20 21:04:29
904	2006	3	2006-Information technology-Computer science-Artificial intelligence	MURAL00022	NULL	2018-04-26 19:37:39
905	1873	3	1873-Information technology-Computer science-Artificial intelligence	MURAL00022	NULL	2018-04-26 19:37:46
906	1868	3	1868-Information technology-Computer science-Artificial intelligence	MURAL00022	NULL	2018-04-26 19:39:17

Figure 5.14 Recommended courses rate table in database

### 5.2.3.2 Collaborative Based Filtering

This section describes the implementation of the collaborative filtering component as described in collaborative filtering section. The collaborative filtering component is implemented using a memory based algorithm. Model based approaches require a training phase and a dataset in order to train the model. However, in the actual stage of this thesis, these requirements cannot be met, and datasets used for scientific and academic studies are created only after costly and challenging processes. Sometimes it can take for years to create a dataset. There are several

algorithms within memory based approaches that are available as explained in chapter 4. For this implementation, ontology similarity and cosine similarity are going to be used in order to calculate the similarity between two users.

As explained in the CBF implementation, users were asked to provide a rating for each course in the recommendation list. CF class is the core component on the server side which has several functions it needs to carry out to provide a recommendation list which is called the KNN list as shown in Fig.5.15.

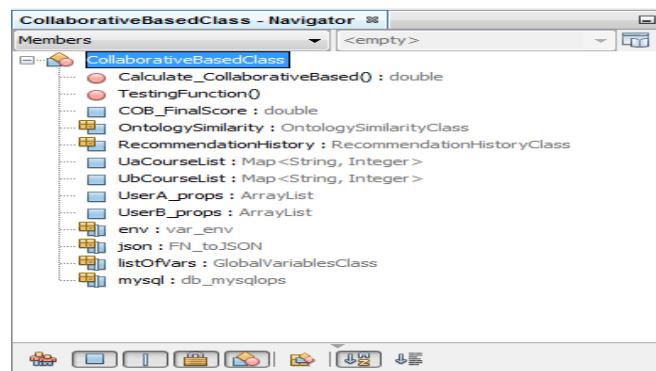


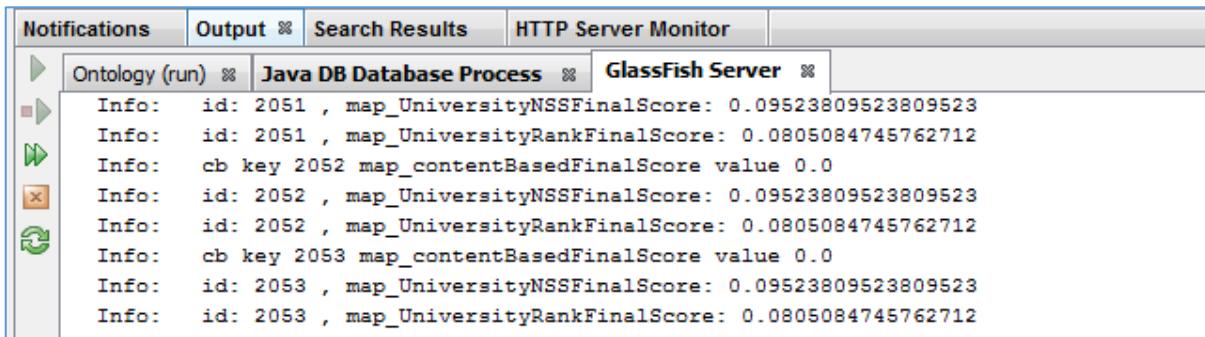
Figure 5.15 Snapshot of main functions in CF classes

The main aim of CF is to find similar users to the target user and to calculate the similarity to find the K-top nearest neighbours and provide the user with courses that have the highest rate. To do so, two classes in code have been implemented which are called *ontologySimilarity*, *RecommendationHistory*. OPCR applied ontology in the CF part in order to obtain a high quality result for users' similarity. As mentioned in chapter 2, one of the main drawbacks of CF is that it needs a large number of rates for the items in order to find similar users to the target user. However, in most cases, users are too lazy to rate an item which was recommended to them. Using ontology to calculate the user profile information helped to overcome the problem of failure to rate an item.

### 5.2.3.3 Final Recommendation List

OPCR used a hybrid approach to recommend items to users as detailed in chapter 4. The implementation process of the final recommendations list combines the recommendation lists from CBF and CF by mapping similar courses in each list and computing the similarity degree for similar courses. According to the final scoring function in Eq. 4.12, the other factors weights such as university rank and NSS score will be added to the course weights in the final list as shown in Fig.5.16. Subsequently, the core function will re-rank the recommended

courses' positions on the list. The top five courses which have a high similarity degree will be displayed to users.



```

Notifications Output Search Results HTTP Server Monitor
Ontology (run) Java DB Database Process GlassFish Server
Info: id: 2051 , map_UniversityNSSFinalScore: 0.09523809523809523
Info: id: 2051 , map_UniversityRankFinalScore: 0.0805084745762712
Info: cb key 2052 map_contentBasedFinalScore value 0.0
Info: id: 2052 , map_UniversityNSSFinalScore: 0.09523809523809523
Info: id: 2052 , map_UniversityRankFinalScore: 0.0805084745762712
Info: cb key 2053 map_contentBasedFinalScore value 0.0
Info: id: 2053 , map_UniversityNSSFinalScore: 0.09523809523809523
Info: id: 2053 , map_UniversityRankFinalScore: 0.0805084745762712

```

Figure 5.16 Snapshot of mapping university rank and NSS score to final scoring function

### 5.3 Summary

This chapter has described the implementation of OPCR framework, the implementation of each OPCR component and the configuration directives which decide the system's behaviour. Data extraction based on ontology has tested to extract course information and job information, the test results showed useful of using an ontology to extract most relevant information. Ontology mapping module in the framework has been tested by using protégé tool. All the algorithms of proposed recommender system OPRCourse have been tested. Results showed OPRCourse able to reduce the information overloading problem. Next chapter described in details the evaluation process for whole framework.

# CHAPTER 6 EXPERIMENTAL EVALUATION

*“If you can’t measure it, you can’t improve it.”*

Peter Drucker

## 6.1 Experimental Study

Unlike in different domains, there are no standard datasets nor standard evaluation methods for evaluating course recommender systems(Drachsler et al., 2008a). This restricts the comparison of evaluation results between course recommender systems. Evaluation frameworks exist to support offline experiments on datasets, for instance for the evaluation of folksonomy-based recommender systems or the simulation of multi-criteria recommender systems (Nikos Manouselis & Costopoulou, 2006), but these solutions have very narrow usage scenarios. Experimental evaluation has been designed based on the OPRC framework. All modules that have been developed use open source tools which have been organised in a traditional client and server structure. The main objective of the evaluation is to determine whether the proposed method, which considers ontology data integration and hierarchically-related concepts, is better than the existing filtering method, which does not consider hierarchically-related concepts.

To achieve the objectives, we organised an experiment in which participants used an experimental system for evaluating course items. We made sure that user interaction with the framework was flexible which allowed the participants to select and rate the items of the university course in several sessions; for example, they could use the CBF and CF algorithm individually to see how the results changed compared with the OPCR algorithms. The participants were asked to provide a rating for each item on the recommendation list and re-rank the position of the item in the recommendation list. The participants’ ratings were then compared with the system’s rankings.

## 6.2 Description of the experiment

The experiment began with requested students from different academic backgrounds from the University of Portsmouth to participate in our framework experiment. A total of 123 students participated in the month-long experiment. The students were from two different departments, the School of Computing and the School of Business and Management. After evaluating the system, the participants were asked to answer questions regarding different aspects of the system’s performance. A total of 95 students responded to the questionnaires, including 50

students from the School of Computing and 45 from the School of Business and Management. The participants were from different levels of education and study, including undergraduate, postgraduate and PhD students. Table 6.1 shows the number of students from each level.

Field of study	Study level	No. students
Computer Sciences	PhD	5
	MSc	15
	BSc	30
Business and Management	PhD	5
	MSc	16
	BSc	24

Table 6.1 number of participants and level of study

Each participant that registered onto the system recommended courses based on his/her profile. The users were asked to give a rating on the recommended courses and re-rank the recommended positions. The participants were also asked to use the search criteria to search on the UCAS website and rank the user satisfaction in both cases. The course dataset used in the experiment was from the UCAS website.

The experimental system used UCAS as the main source for course information on each day of the experiment. We collected all of the course items by using the web crawler that had been built and customised to extract course information. Each user was required to rate all the course items that were in the recommendation list provided on the day of the experiment.

Each participant could use the system in two ways; one option was a general search based on keywords, and the second was a personalised search achieved by building a user profile. This is undertaken by the register in the system and gives the system information about the user's educational background and interests. After the user profile has been built, the system search will become more personalised. The system will recommend the five top courses to participants that are more relevant to their user profile. The participant is required to rate each course that is on the recommendation list, based on their interest in it, on a scale of 0 (not interested at all) to 5 (strongly interesting). Several recommender systems use a 1-5 scale, particularly course filtering systems and news filtering systems, such as NewsWeeder (Lang, 1995) and the commercial Amazon system (Linden et al., 2003). The final course recommendation list showed that each participant used CBF and then CF, as shown in Fig.10. The participants

registered and each defined an initial profile. The initial profile consisted of two main parts; the first was personal information, such as username, gender, postal address, user contact. The second part included academic information for the user, such as field of study, main subject, major subject, current study level, interest areas, course language preferred and skills.

Each user's profile was updated implicitly by giving consideration to the course that was rated in the recommendation list by the user. The weight of each level was increased if the user rated the item relatively highly. The degree of the relevance of the recommended items was adjusted by using a certain threshold of the rating range.

Each participant used the system three times in order to create different profiles with a different search. The participants' user profiles were updated by the data collected from the experimental system. This data was also used in different variations of the algorithm's runs. The system's performance was evaluated against a ranked list of the items, as rated by the participants.

Several metrics have been used to analyse the results that were collected during this experiment. The users' ratings on the 0-5 scale were saved so as to enable a ranking order of the courses, and thereby express the items' relevance to the user. The questionnaire was used to measure user satisfaction and the quality of the recommendations. A benchmark was used to compare OPCR with the current system.

The screenshot shows a web interface for course and job recommendations. At the top, there is a search bar with a red 'SEARCH' button. Below the search bar, the text 'We got courses recommended for you' is displayed. A table lists three recommended courses:

Course ID	Course Name	Field of Study	Major subject	University Name	Course Fee	Course Duration	Course Location	Course URL	Rate
2054	Artificial Intelligence	Information technology/	Artificial intelligence	University of Edinburgh	4,370	3 years	Central area campus - EH8 9JS	<a href="#">course website</a>	<input type="button" value="▼"/>
1954	Data Science and Computational Intelligence	Information technology/	Artificial intelligence	Coventry University	7,750	1 year	Main Site - CV1 5FB	<a href="#">course website</a>	<input type="button" value="▼"/>
2125	Computer Science	Information technology/	Artificial intelligence	Nottingham Trent	9,185	2 years	Clifton Campus -	<a href="#">course website</a>	<input type="button" value="▼"/>

Below the course section, the text 'We got jobs recommended for your course major' is displayed. A message says 'This may take some time to get all jobs please be patient'. There is a filter bar with 'Filter by location,Job title,salary...' and a note 'Click "Job Details" for more details'. A table lists recommended jobs:

Job Details	Job ID	Job Title	Company	Salary	Reviews
<input type="button" value="Job Details"/>	J18126	Software Developer ( Artificial Intelligence )	C4Commerce London	£35,000 - £45,000 a year	50 reviews

Figure 6.1 Course and job recommendations

## **6.3 Data Source and Configuration**

In this thesis, the data collection of the content of MSc courses was gathered from the UCAS (Universities and Colleges Admissions Service) website, and Indeed.com website was then used for job information as mentioned in section 3.3.1. In order to achieve this, a web crawler was built and customised. The collected data was used to construct the item ontology (courses and jobs), based on our knowledge. There was no existing dataset for master's courses at UK universities. We have created our dataset, called ontologyset, which includes courses extracted from UCAS.com. However, there was no need for an established benchmark dataset to evaluate OPCR's performance. The system metadata included close to 21,000 online courses in ontologyset, covering 70 diverse subject areas that had been archived from UCAS.com. These were focus chosen and downloaded from different departments at various universities and colleges in the United Kingdom for testing purposes. The breakdown was to select 20 of these subject areas with a number of courses. However we decided to use the computer sciences and business management courses. Courses in ontologyset cover every postgraduate academic level, which yields a representative set that includes a wide range of courses offered at different universities.

We used the Indeed.com website as a source in order to extract job information. This information included the job title, description, salary, location and user reviews. For test purposes, any jobs related to CS and BAM courses were extracted.

## **6.4 Evaluation Metrics**

There are many approaches to evaluating the recommendation systems. The evaluation can use either offline analysis or online user experimental methods or a combination of these two approaches (Herlocker, Konstan, Terveen, & Riedl, 2004). The RS aims to offer choice preferences associated with the context of the domain, as well as to be undividedly on user preference and interests. However, user satisfaction can fluctuate according to what the user needs to achieve. The approaches will be discussed in detail in the following subsections.

### **6.4.1 Offline Evaluation**

An offline evaluation is achieved by using a pre-gathered dataset of users who choose or rate items. In many cases, the offline evaluation will be useful as it will enable knowledge about user behaviour to be obtained, such as the movie domain and music domain (Shani & Gunawardana, 2011). However, it will be difficult to obtain accurate results for the user's

interests in the education domain because each user needs to choose a different education path based on their preferences. For this reason, the online evaluation obtained more accurate results because it was possible to obtain a real user interaction with the recommendation system.

#### 6.4.2 Online evaluation

In an online evaluation, users interact with a running recommender system and receive a recommendation. Feedback from the users is then collected by either questioning them or observing them. Such a live user experiment may be controlled (e.g. randomly assigning users to different conditions) or a field study may be used in which a recommender system is deployed in real life, and the effects of the system then observed. Online evaluation is the most desirable as it can provide accurate results of how effective our system is with real users (Beel & Langer, 2015). Conducting such evaluations is both time-consuming and complicated, but it is inevitable that we must conduct an online evaluation for this research since it is the only way to measure real user satisfaction. Their multiple metrics have been used to evaluate factors, such as recovery, the accuracy of relevance and rank accuracy, as follows.

##### 6.4.2.1 Recovery

The recovery metric has been employed to evaluate how the recommender algorithms performed in providing a proper ranking to the whole item set (Hernández del Olmo & Gaudioso, 2008). The user prefers a kind of system that provides a higher rank for items which are relevant to the target user. Items that are relevant to each user can be extracted, based on her/his ratings in the test dataset. We considered the course selected by a test user and found that the Like rating (ratings 3, 4, 5) in the test dataset was relevant to the active user. Therefore, the recovery  $RC$  can be obtained according to Eq. (6.1):

$$RC = \frac{\sum_{u \in u_{\text{TestSet}}} \frac{1}{Ku} \sum_{i=1}^{Ku} \frac{pi}{Cu}}{|u_{\text{TestSet}}|} \quad (6.1)$$

Where  $Cu$  is the number of candidate items for a recommendation in an item set,  $Ku$  is the number of relevant items to user  $u$ ,  $pi$  is the place for an item  $I$  in the ranked list for user  $u$ , and  $|u_{\text{TestSet}}|$  is the number of users in the test dataset. Based on this definition of recovery, the lower the  $RC$  is, the more accurate the system. In Table 6.2, an example of measure recovery metric, five users received a list of recommended courses and they rated (R) these according to their individual needs. We used Eq. 13 to find the recovery metric value as following:

Ranking list	User 1		User 2		User 3		User 4		User 5	
	Course	Rate								
1	C1001	4	C1022	1	C1001	3	C1066	5	C1066	2
2	C1004	5	C1034	4	C1222	2	C1032	5	C1032	5
3	C1012	2	C1012	5	C1432	3	C1032	2	C1032	2
4	C1023	2	C1023	2	C1004	4	C1033	2	C1033	2
5	C1009	3	C1055	2	C1012	1	C1012	1	C1012	1

Table 6.2 Example of Recovery Metric

$$R = \frac{\frac{1}{3}(\frac{1}{5} + \frac{2}{5} + \frac{5}{5}) + \frac{1}{2}(\frac{2}{5} + \frac{3}{5}) + \frac{1}{3}(\frac{1}{5} + \frac{3}{5} + \frac{4}{5}) + \frac{1}{2}(\frac{1}{5} + \frac{2}{5}) + \frac{1}{1}(\frac{2}{5})}{5} = 0.45$$

The threshold of relevance course in the example above is  $\geq 3$ , R value change according to the number of relevance item that have been rated with value  $\geq 3$ , as we mentioned rate scale range is (1-5). In table 6.2 relevant courses is 11 courses from the total 25 courses. The R value will increase when number of relevant increase. In our experiment  $|u_{TestSet}|$  is 95 participants and the average of R value that we obtained is 0.36 and that is refer to accurate of the recommendations list that provided by proposed recommender system in OPCR.

#### 6.4.2.2 Accuracy of List Relevance

In an ideal information retrieval system, documents should be ranked in order of how probable their relevance or usefulness is. Most IR and RS follow this principle and will be presented to the user in a list. There are several methods that have been presented in the past which measure the accuracy of the relevance. One of these methods is average precision (AP) (Hernández del Olmo & Gaudioso, 2008). This is the average of the precision value that is obtained from the set of top k documents that exist after each relevant document is retrieved for the single query (for one recommendation list). If we have a set of queries (many recommendation lists), then we need to determine the mean average precision MAP as shown in Eq. (6.2) and Eq. (6.3).

$$Average\ Precision\ (AP) = \frac{1}{M} \sum_{i=1}^n rel(k) \times P_{rec}@k \quad (6.2)$$

$$mean Average Precision (MAP) = \frac{1}{M} \sum_m AP_m \quad (6.3)$$

Where

M: the total number of relevant documents

n: The list length

rel (k): 1 if relevant, otherwise 0

$P_{rec}@k$  : precision at rate 3 and above at each rank

m: number of queries

According to the example in Table 6.3, we have five users who received a list of recommended courses and they rated (R) this based on their interest. We used Eq. (6.2) to obtain the average precision for each user as the following:

User 1		User 2		User 3		User4		User 5	
Course Rec. List	Rate								
C1001	4	C1022	1	C1001	3	C1066	5	C1066	5
C1004	5	C1034	4	C1222	2	C1032	5	C1032	2
C1012	2	C1012	5	C1432	3	C1032	2	C1032	2
C1023	2	C1023	2	C1004	4	C1033	2	C1033	2
C1009	3	C1055	2	C1012	1	C1012	1	C1012	1

Table 6.3 Example of Accuracy of list relevance

$$average percision for user 1 = \frac{\frac{1+2+3}{1+2+5}}{3} = 0.86$$

$$average percision for user 2 = \frac{\frac{1+2}{2+3}}{2} = 0.58$$

$$average percision for user 3 = \frac{\frac{1+2+3}{1+3+4}}{3} = 0.8$$

$$average percision for user 4 = \frac{\frac{1+2}{1+2}}{2} = 1$$

$$average percision for user 5 = \frac{\frac{1}{1}}{1} = 1$$

Many applications have been designed so that they recommend  $N$  items to users. Precision for the list recommended user  $u$ ,  $P_u(N)$  is defined as the percentage of the relevant items to user  $u$  in the list recommended to the user. We considered items selected by the target user in the test dataset and received *Like* rating (such as 3,4,5) as relevant items to the target user. The precision of the systems on a recommendation list with  $N$  items can be defined in Eq. (6.4) as:

$$P(N) = \frac{\sum_{u \in u_{\text{TestSet}}} P_u(N)}{|u_{\text{TestSet}}|} \quad (6.4)$$

According to the example in Table 6.4, the precision will be as the following:

$$P(N) = \frac{\frac{4}{5} + \frac{3}{5} + \frac{4}{5} + \frac{3}{5} + \frac{3}{5}}{5} = 0.68$$

User 1		User 2		User 3		User 4		User 5	
Course Rec. List	Rate								
C1001	4	C1022	3	C1001	3	C1066	5	C1066	5
C1004	5	C1034	4	C1222	5	C1032	5	C1032	2
C1012	3	C1012	5	C1432	3	C1032	2	C1032	4
C1023	2	C1023	2	C1004	4	C1033	3	C1033	4
C1009	3	C1055	2	C1012	1	C1012	1	C1012	1

Table 6.4 Example of the percentage of relevant items to user  $u$

In our experiment the average of  $P$  value that we obtained for 95 participates is 0.78 and that is refer to accurate of the recommendations list that provided by proposed recommender system in OPCR.

#### 6.4.2.3 Rank Accuracy

Rank metrics extend recall and precision to take the positions of correct items in a ranked list into account and measure the ability of an algorithm to produce an ordered list of items that match the opinion of the user. Relevant items are more useful when they appear earlier in the recommendation list than when the item appears at the bottom of the list and are particularly important in recommender systems as lower ranked items may be overlooked by users. We used the Spearman's ranking correlation  $r$  to calculate the ranking metric for the system (Shani

& Gunawardana, 2011). The ranking will be more accurate when the r value is close to (1). For the calculation method of Spearman's ranking correlation we used Eq. (6.5)

$$r = 1 - \frac{6}{n(n^2 - 1)} \sum_{i=1}^n (x_i - y_i)^2 \quad (6.5)$$

Where n is number of recommended items

$x_i$  is the rank of item i output by RS

$y_i$  is the rank of item i offered by the user

In order to explain how to measure rank metrics we have two cases scenarios, the example of the first case is shown in Table 6 for user U1, all the user rank is different from the system rank. We used Eq. (17) to find the value of rank metrics as the following:

Recommendation courses for U1	User rate	System rank	User rank
C1001	4	1	2
C1004	5	2	1
C1012	2	3	5
C1023	2	4	3
C1009	3	5	4

Table 6. 5 Example of Represent System Ranking and User Ranking case1

$$r = 1 - \frac{6}{5((5)^2 - 1)} ((1 - 2)^2 + (2 - 1)^2 + (3 - 5)^2 + (4 - 3)^2 + (5 - 4)^2)$$

$$r = 0.6$$

The second case is for user U2 as shown in Table 6.6. In this case we noticed that 3 over 5 recommendation ranks are similar in both the system and user ranking and when implemented with Eq.17 the result will be as following:

$$r = 1 - \frac{6}{5((5)^2 - 1)} ((1 - 2)^2 + (2 - 1)^2 + (3 - 3)^2 + (4 - 4)^2 + (5 - 5)^2)$$

$$r = 0.9$$

Recommendation courses for U2	User rate	System rank	User rank
C1001	3	1	2
C1004	5	2	1
C1012	2	3	3
C1023	1	4	4
C1009	1	5	5

Table 6.6 Example of Represent System Ranking and User Ranking case2

Furthermore, to measure the ranking metric for all the users, it is necessary to calculate the average for all the r-value for the testing users. All the participants in the evaluation have asked to test the recommendation list and rank the recommended course base on their individual interests. Based on the results, the average Spearmans's ranking corrections across all the 95 students, with using ontology and without using ontology model, were 0.8534 and 0.6732. This indicates that the recommendation given where using ontology similarity was providing the best results.

## 6.5 Experimental Results

The experimental data that we collected, i.e. the user ratings, was used to both train and test the hybrid filtering algorithms with the ontology technique. We implemented OPCR in Java and ran it on an Intel(R) Core(TM)2 Dup CPU processor, with a CPU of 3.20 GHz and 16 GB of RAM, under Windows 7. HTML was used for the system interface, and the MySQL server was used to allocate a system dataset and user rating. In addition, a protégé tool was used to evaluate the ontologies built into the system.

The effectiveness of OPCR was assessed in an empirical study that used a group of university students who played the role of appraisers at our university in order to evaluate the performance of OPCR. To recruit the appraisers, they were firstly asked to create their user profile and verify the usefulness of the recommended courses. We presented our empirical study to two departments at the University of Portsmouth, CS (Computer Sciences) and BAM (Business and Management). Since these participants differed in their majors and their academic standing, they formed a group of diverse appraisers. Altogether, 123 appraisers were recruited which represented a range of groups, from undergraduate to postgraduate level, across 37 different

majors. Additionally, each appraiser was asked to modify his/her profile twice during the evaluation process so that different courses would be requested with each modification. This produced a yield close to 200 cases that were used to verify the performance of OPCR.

The three performance measure metrics mentioned in the online evaluation section were used to evaluate the results obtained from the participants in order to make a comparison between the traditional CBF and CF filtering algorithms and the OPCR algorithms. The objective of this experiment is to evaluate in which circumstances and how the ontology-based learning algorithm and the CBF-CF recommendation algorithms enhanced with semantic information improve the performance of the recommendation system in terms of recommendation quality and accuracy. To evaluate this, two different configurations of the recommender have been set up:

- The OPCR, which is the configuration working with the ontology model components developed in this work and presented in chapter 3;
- The Rec, which is the same recommender but without exploiting the ontology-based components: the weighting propagation based on domain inferences in the learning algorithm, and the keyword-based similarity measure and the find top similar users based on rating in the recommendation algorithm.

In this experiment the recommendation algorithm of the two configurations are compared: one is the algorithm presented in this work with all the ontology based components activated (OPCR), and the other is the same algorithm but with the keyword based similarity measure and the find top similar users based on rating (Rec). The average values for each metric are presented in Fig.6.2, was that the proposed approach algorithms worked far more precisely than the traditional one.

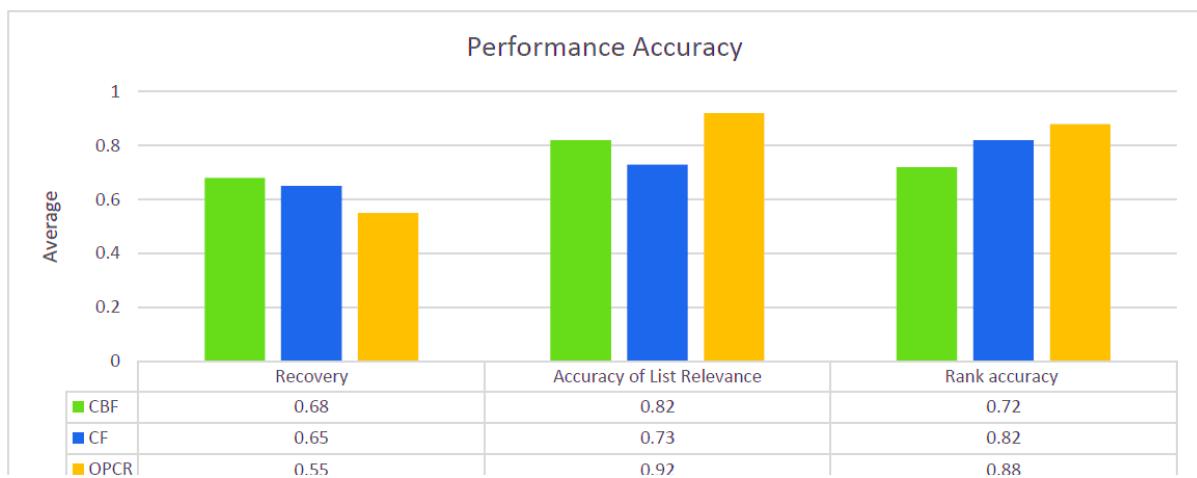


Figure 6.2 Comparison between OPCR and (CBF, CF) performance metrics

From the above results, the conclusion can be obtained about the type of improvement achieved using the hybrid recommendation algorithm enhanced with the ontology similarity, is that the OPCRa configuration obtained, on average, the best result is an indicator that the reduction of the number of relevant courses is considered as an improvement in accuracy, since the reducing the set of possible courses to recommend.

Moreover, OPCR has been compared with some current course finder systems, such as UCAS, it showed that OPCR is more accurate and provides more personalised results than UCAS. The performance was also of a higher quality than that provided by UCAS, as shown in the Fig.6.3.

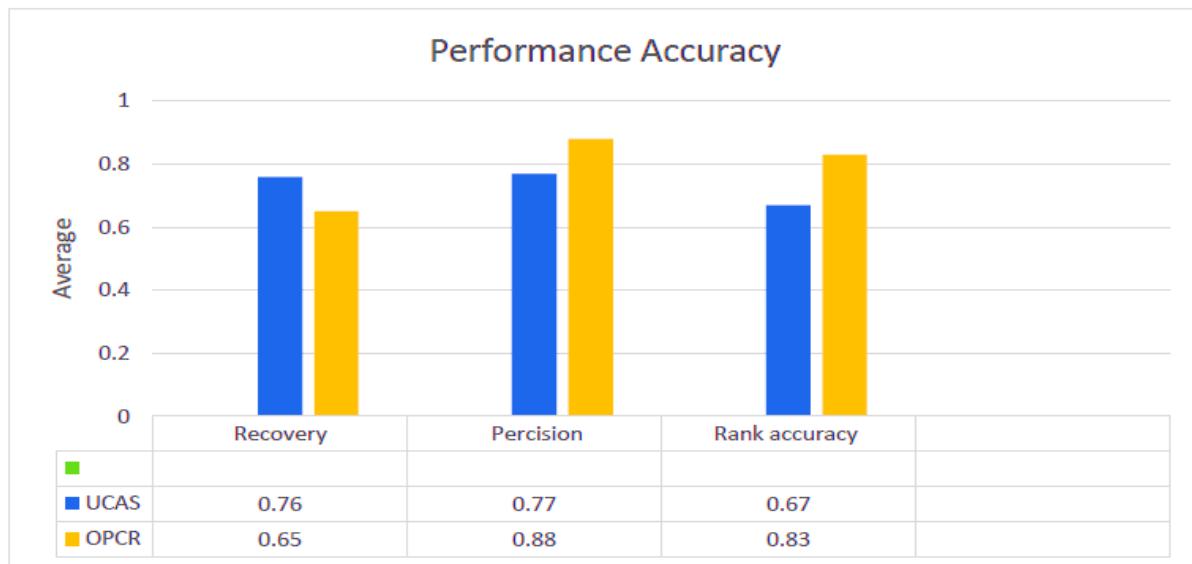


Figure 6.3. Comparison between POCR and UCAS performance metrics

In contrast, we used a questionnaire (as shown in Appendix D) to evaluate both user satisfaction and the quality of the items recommended to the participants. The questions were designed according to the design guidelines and principles, and are described in more detail by (Brace, 2005). The Likert-type scales used statements such as: "Please rate the extent to which you agree/disagree with the following", and 5-point response scales have been used. The response scales used anchors such as 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree, as shown in Fig. 6.4. The sample of the questions was as follows:

*Q1 Overall, I am satisfied with this recommender system.*

*Q2 I am convinced of the items recommended to me.*

*Q3 I am confident I will like the items recommended to me.*

*Q4 This recommender system made me more confident about my selection/decision.*

*Q5 The recommended items made me confused about my choice.*

*Q6 This recommender system can be trusted.*

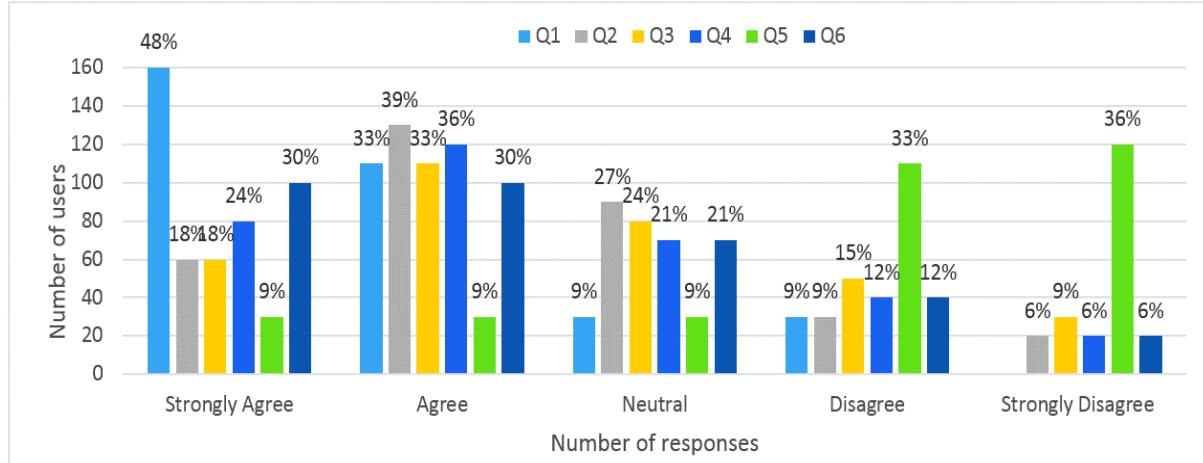


Figure 6.4. Attitudes Questions

The results showed that 81% of the participants were satisfied with the recommendations they received. Ontology-based recommendations helped the users to obtain a more suitable recommendation. Moreover, 66% of the participants agreed that the recommendation system had helped them to make the right decision without making them feel confused about what was an appropriate choice. We have considered many of the other factors that are required to obtain an accurate result regarding the quality of the recommended item, and the user satisfaction of the OPCR as follows:

## 1. Quality of Recommended Items

### 1.1 Accuracy

Questions regarding accuracy evaluated how likely it was that users would see that the course recommended to them matched their interest (e.g. the location of the university, the financial budget). The second question about the accuracy measurement was whether the system recommended good suggestions that would help with the decision-making process. The accuracy questions were as follows:

*Q1 The items recommended to me matched my interests*

*Q2 This recommender system gave me good suggestions*

The results are shown in Fig.6.5, and more than 60% of the users were satisfied with the recommended courses regarding how the recommended course matched with users' interests.

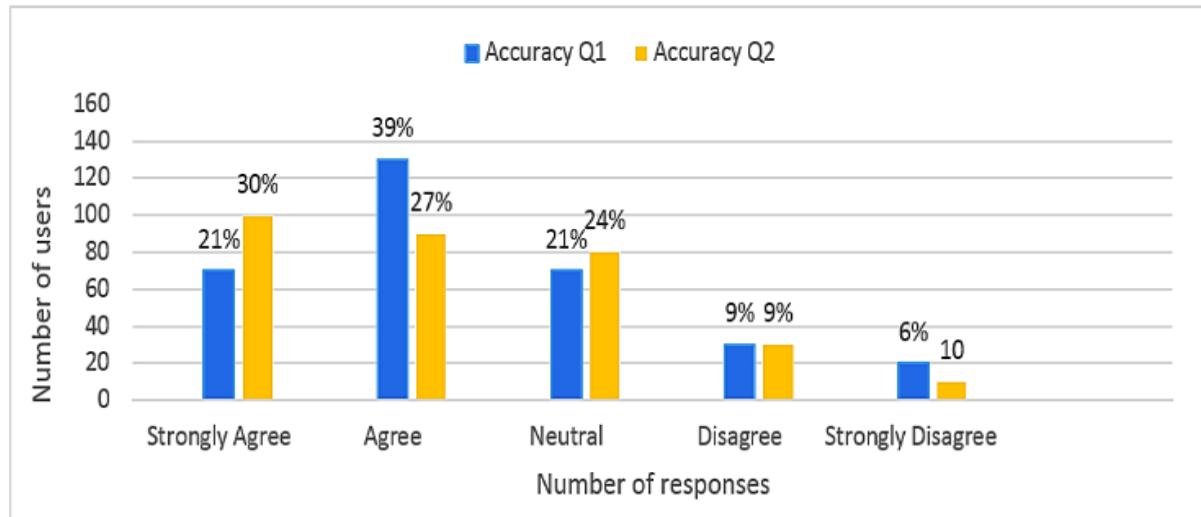


Figure 6.5. Accuracy Questions

## 1.2. Familiarity

Familiarity captures how well the users know some of the recommended items. OPCR used an ontology-based recommendation technique to recommend the most relevant items to users. The users were asked, “*are some of the recommended items familiar to you?*” The responses showed that 24% of the users had obtained recommendations which included some familiar items, and 33% of the users said the results included new items, as shown in Fig.6.6.

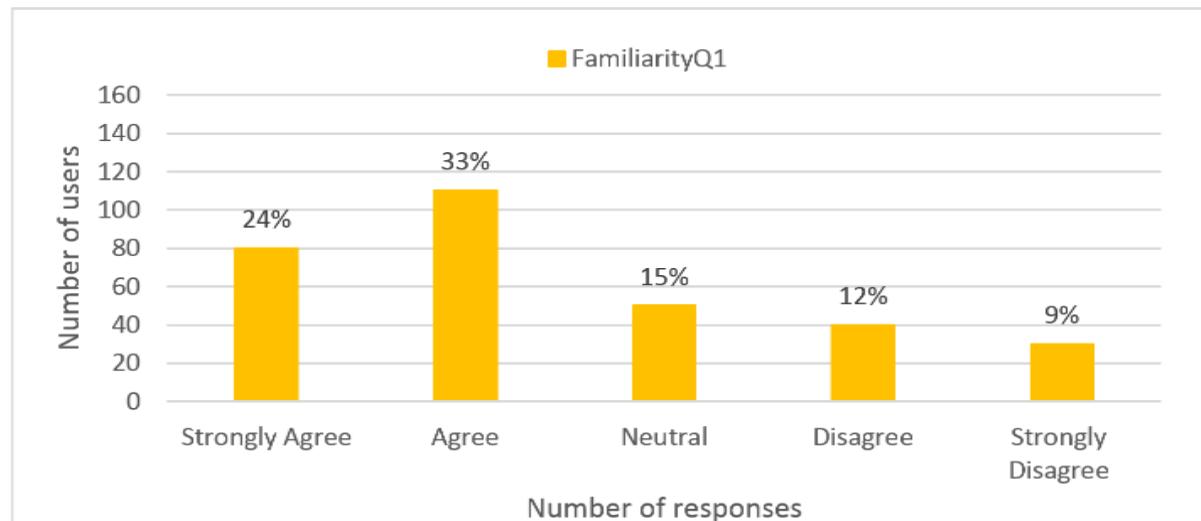


Figure 6.6. Familiarity Question

### 1.3. Novelty

Novelty is one of the important indicators of user satisfaction as it helps users in the decision-making process(Liang Zhang, 2013). OPCR provided the users with recommendations that included novel items which were not expected because ontology mapping is able to link all of the attributes in the course profiles and user profiles. Recommendations were included for novel items and also helped the user to discover new items, according to the results of the user's responses to the novelty questions below 60% of users have found a novel item in the recommendation list and 54% of the users have discovered a new item which they were not expected as shown in Fig.6.7,

*Q1 The items recommended to me are novel*

*Q2 This recommender system helped me discover a new course*

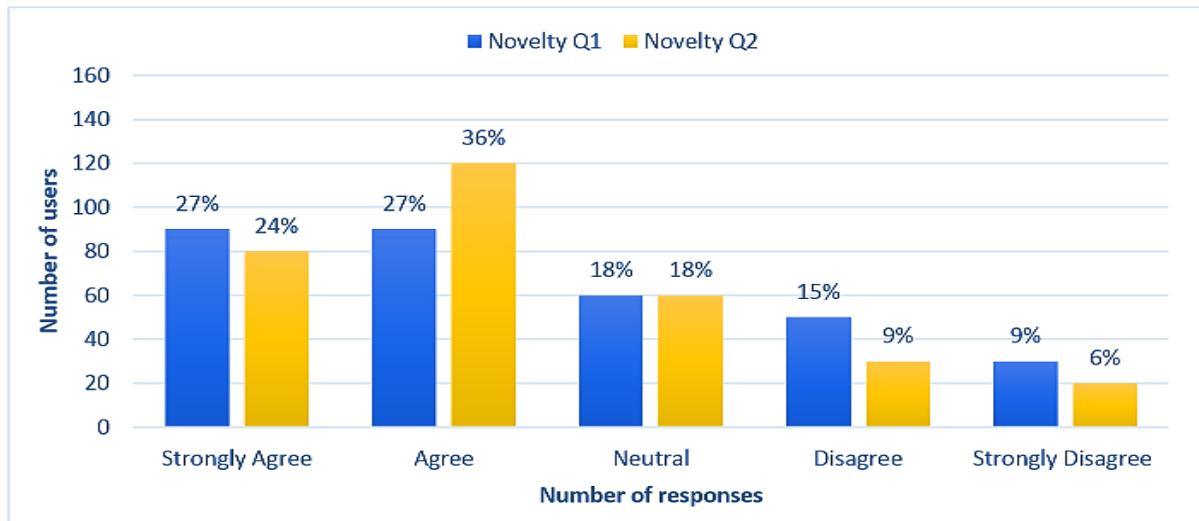


Figure 6.7. Novelty Questions

### 1.4. Diversity

The course domain in the recommendation system is different from that of other domains, such as news and movies (Parameswaran, Venetis, & Garcia-Molina, 2011). OPCR mainly recommended courses based on content-based filtering, which measures the similarity between the user profile and the item. The recommendations are similar to each other because the ontology mapping technique will not allow irrelative items to appear with the recommendation items. We asked two questions to understand whether the recommendations had diverse items and how similar the recommended items were to each other. The results in Fig.6.8 show that more than 65% of the recommendations items have no diversity.

*Q1 The items recommended to me are diverse.*

*Q2 The items recommended to me are similar to each other*

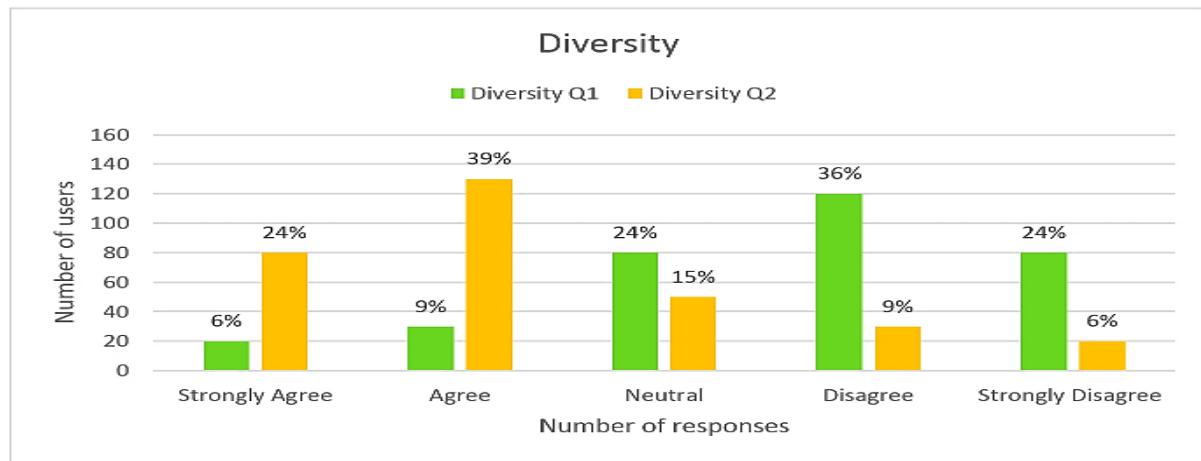


Figure 6.8. Diversity Questions

## 2. Interaction Adequacy

OPCR is a flexible system that can dynamically modify any part that is related to the recommender engine or user profile. The user can give a rating for the recommended course, with the scale of the rating adjusted from (1-5). To measure how interactive, the system is with the user and how satisfied the user is with the user interface, the users were asked the following questions. The results are shown in Fig.6.9.

*Q1 This recommender system allows me to tell what I like/dislike.*

*Q2 This recommender system allows me to modify my taste profile.*

*Q3 This recommender system explains why the courses have recommended to me.*

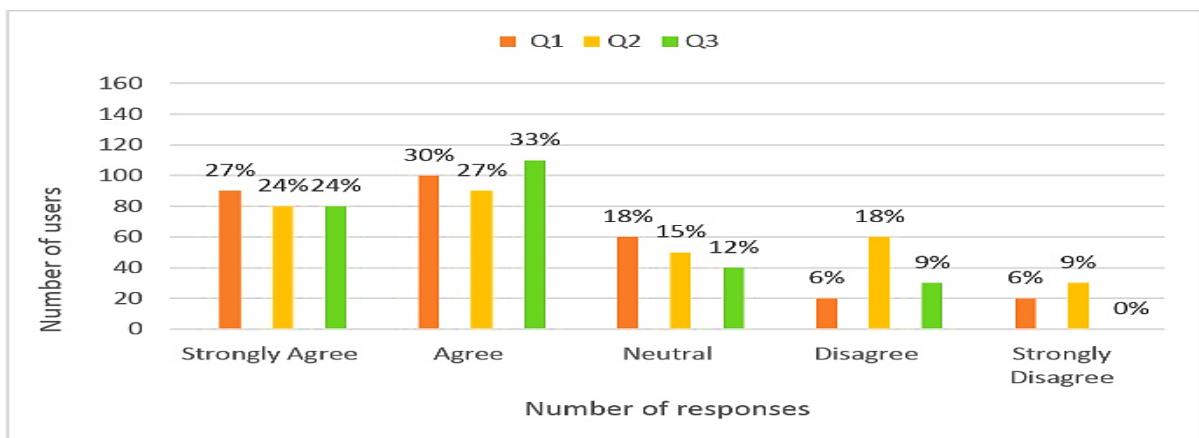


Figure 6.9. Interaction Adequacy Questions

### 3. Perceived Ease of Use

The term of perceived ease of use emphasizes the decision efficiency. (Pu & Chen, 2010) From the model of Technology Acceptance Model (TAM), perceived ease of use is salient factor that affect the intention of adoption. Therefore, to evaluate whether a travel recommender system elicits users preference or not can help to estimate users intention of adoption. We have asked participants two questions in order to know the degree of user satisfaction about using OPCR, the question as the following and the results shown in Fig.6.10:

*Q1 I became familiar with this recommender system very quickly.*

*Q2 I easily found the recommended items.*

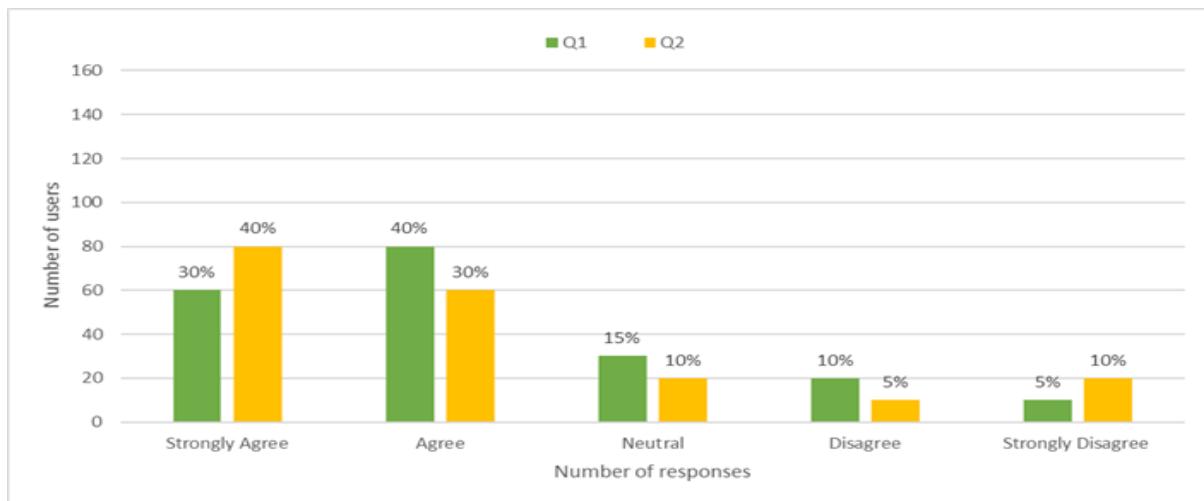


Figure 6. 10. Perceived Ease of Use Questions

### 4. Perceived Usefulness

Rong (R. Hu, 2010) studied into users, the perception of efficiency and accuracy by comparing those who used and didn't use the recommender systems. The term of usefulness mentioned in their work emphasises the decision support and decision quality. We can predict users' future intention of selection via examining whether a course recommender system support users' decision and achieve their needs. All participates were asked the questions in the below to understand and perceived usefulness of OPCR. The result in Fig 6.11 shown that most of the user found the proposed system is very useful.

*Q1 This recommender system helped me find the ideal item.*

*Q2 This recommender system influenced my selection of items.*

*Q3 Using this recommender system to find what I like is easy.*

*Q4 Finding an item to course select with the help of this recommender system is easy.*

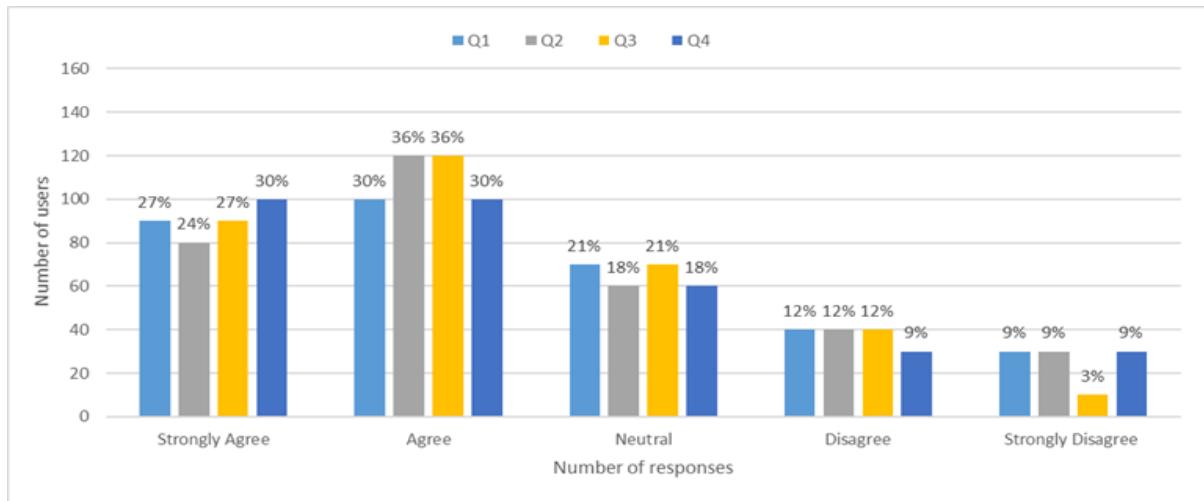


Figure 6.11. Perceived Usefulness Questions

## 5. Control Transparency

According to (Sinha & Swearingen, 2002) users also like to know why an item was recommended even for items they already liked. This suggests that users are not just looking for blind recommendations from a system, but are also looking for a justification of the system's choice. OPCR allows the user to provide the recommendation system to be their personal information and academic information, which is helped to personalised the recommendation to them. Users can go in an easy way able to manage their profiles and update the information. The participants have answered the follows equations to describe their feedback of control transparency of the framework. User has asked for response five questions to measure the user satisfaction of control transparency in OPCR as follows:

*Q1 I feel in control of telling this recommender system what I want.*

*Q2 I feel in control of modifying my taste profile.*

*Q3 Using this recommender system to find what I like is easy.*

*Q4 I understood why the items were recommended to me.*

*Q5 This system's recommender system seems to control my decision process rather*

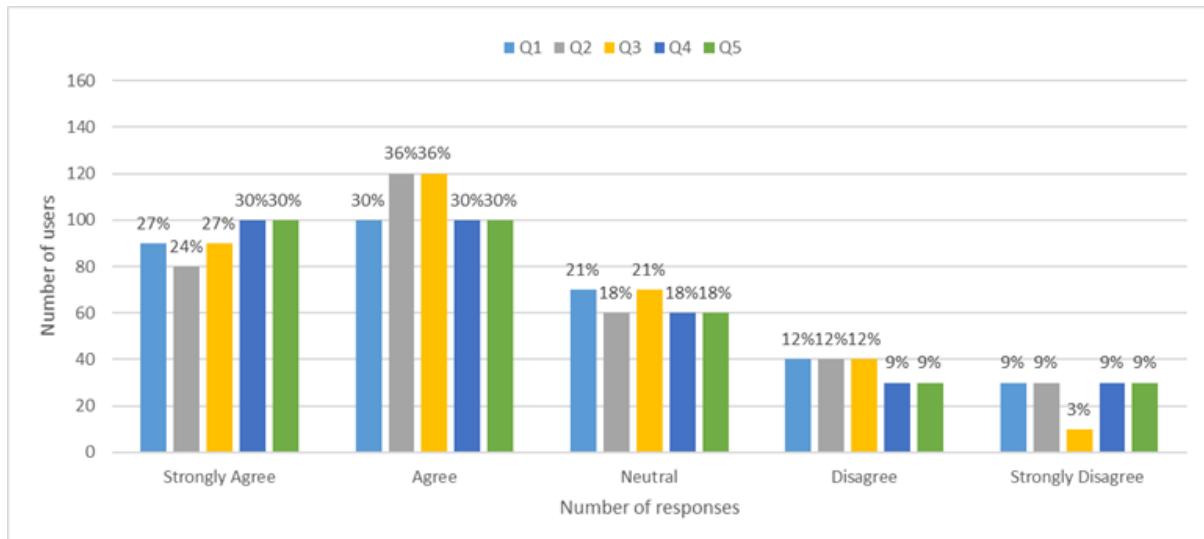


Figure 6.12. Control Transparency Questions

Figure 6.12 shows more 80 % of the users were found that OPCR is easy to use and they were satisfied with the recommendations that provide to them.

## 6.6 Summary

In this chapter, the experimental evaluation configuration and information sources have been described. Furthermore, the main recommendation system evaluation approaches have been identified. According to the domain of the recommended items, which are university courses, the offline evaluation approach failed to provide accurate results regarding the performance of the recommendation algorithms. The online evaluation approach proved to be more suitable to evaluate the framework performances which was achieved by asking 123 students from the University of Portsmouth to participate in an online test of the framework. The participants were requested to test the framework in different situations, with and without using the ontology model with the recommender system. After each test, the participants were asked to respond to the questionnaires regarding the performance of OPCR from different perspectives as shown in Appendix D.

The evaluation results demonstrate the efficacy of OPCR since more than 76% of participants were satisfied with the recommendation list and the results met with their individual needs. Equally, they found that using the ontology model with the recommender system gave them more accurate results because the ontology model seeks to overcome the new user cold start problem by measuring ontology similarity between the user despite the fact that the target user had not previously provided rating information.

# CHAPTER 7 CONCLUSION AND FUTURE WORK

This chapter represents the conclusion to the research path undertaken in this thesis. It provides a brief recap of the main topics that the thesis focuses on and summarises the major contributions provided by this thesis. Certainly, the investigation cannot be considered as complete since much potential remains and therefore the final section outlines some possible research directions worth investigating in the future.

## 7.1 Conclusion

This thesis deals with the issue of information overloading in the education domain. As has been discussed, choosing a higher education course at university can be incredibly tedious and extremely complicated for students. Helping students to make the correct choice from a myriad of available courses in order to meet their individual needs is a real challenge. The existing methods that are mainly based on keywords fail to provide comprehensive knowledge about the course and fail to address the individual user's needs in the recommendation process.

This thesis focuses on the following three main aspects:

1. Create a novel recommendation system framework that can reduce the information-overloading problem and can restrict the amount of options available to the student to fewer relevant alternatives.
2. Propose a new approach for data extraction and integration from multiple sources based on ontology mapping relevant information regarding the recommended items in order to achieve comprehensive knowledge about the recommended items.
3. Design and implement a hybrid recommender system which combines content based filtering and collaborative based filtering utilising ontology to add a significant contribution to overcome the user cold start problem that conventional approaches suffer from. Using ontology similarity with rating values in the collaborative filtering can enhance the ability of the KNN classifier algorithm to find the top nearest neighbour of the target user.

The main contribution of the thesis is that it offers a pathway to an automated and personalised course recommendation system to reduce the information overloading problem through the building of a comprehensive framework which supports data extraction and integration from multiple sources. However, OPCR framework utilises ontology to enhance the recommendation filtering algorithm to deal with the new user cold start problem and to improve performance.

Specifically, this work has made the following contributions:

1. It contributes to the knowledge of existing recommender systems by adding insight as to how the problem is usually tackled and why there remain shortcomings. From a scientific point of view, it makes relevant contributions in the emerging area of ontology-based recommender systems.
2. The creation of a novel recommendation framework based on a dynamic combination of Artificial Intelligence techniques, including collaborative filtering and content-based recommendations supported by the ontology. The framework includes automatic data extraction, integration and personalised course recommendations to provide students with suitable recommendations that meet with their needs. A set of hybrid recommendation algorithms are developed. It not only increases the precision metrics but also reduces information overloading.
3. The ontology model designed to extract and integrate information from multiple sources will contribute to improving the recommendation quality by overcoming the heterogeneity of course information. Also, it features properties, such as generality, which enable it to be used in different recommendation system domains which change with the user's interests and the object's attributes.
4. A personalised recommendation system has been developed and evaluated. The system is available online as open access for researchers and developers.

Data gathering about the items and the user plays an essential role in recommendation systems. In this thesis, a new approach has been proposed to extract the data from multiple sources based on ontology. For course information and career information, an ontology-based crawler has been developed which retrieves web pages according to relevance and which discards the irrelevant web pages using an algorithm. In this approach, a concept of ontology provides a similarity calculation of levels of the concepts in the ontology and the user query, and the relationship between these was used. It is therefore intended that this crawler will not only be useful in exploiting fewer web pages, such that only relevant pages are retrieved, but will also be an important component of the Semantic Web, an emerging concept for future technology. The evaluation results show that the ontology based crawler offers higher performance than that of a tradition crawler. This improved crawler can also be applied to areas such as recruitment portals, online music libraries and so forth.

The aggregation of ontology domain knowledge into the recommendation process is one of the solutions that can overcome the limitations of conventional recommender systems. The ontology mapping algorithm proposed in the framework aims to link the information from multiple sources. The evaluation has shown that this mapping approach improved the performance of information retrieval.

Previous algorithms utilised for course recommendation have many limitations as discussed in chapter 2. Therefore, a hybrid recommendation system was proposed which combines content based filtering and collaborative based filtering utilising ontology. Ontology similarity was used between the course profile and the student profile and also applied to find similar users to the target user. Using ontology in collaborative filtering enhanced the KNN algorithm and improved the cold start problem for a new user because, even though the user has not rated any course, the system is able to find the top nearest neighbour by calculating the ontology similarity from the personal information and academic information held in the student profile.

Finally, based on the proposed OPCR framework and algorithms, a personalised course recommendation system (OPRCoure) has been implemented and evaluated. The experiment results show that the framework in this research can reduce information overloading and provide relevant course recommendations based on individual preferences. The results show that overall student satisfaction was 86% using the ontology model and 56% without using the ontology model. This indicates that the proposed framework is able to offer improved recommendation accuracy and, consequently, user satisfaction.

## 7.2 Limitations

At the current time, the process is not fully automated as the weights of each member of the algorithms need to be decided upon by the developer. Even though not explicitly proven for OPCR, all relevant literature suggests that reducing the number of alternatives is a solution to information overloading. Unlike other recommender systems (e.g. collaborative filtering), the limitation of courses to just a few is not arbitrary, such as the top five better scoring courses, but is based on specific criteria that are understandable and comparable by the student. This is most important since, in some situations, the student will still feel the need to examine the other options that are why trust and transparency are highlighted in the literature as important challenges in the adoption of a new recommender system.

The use of ontology for knowledge representation in ontology-based recommenders for an education domain has the potential to improve the quality of recommendations. This

experiment and the evaluation evidenced in this thesis reveal that there has been significant growth in research on the ontology-based recommendation. There are, however, still certain challenges to be faced by researchers in this field.

Firstly, creating ontologies is a difficult and time-consuming process that requires skill in knowledge engineering. Furthermore, the acquisition of ontology knowledge in the context of the education domain requires expertise in this area. Secondly, the evaluation of ontology-based recommenders for the education domain is another challenge experienced by many researchers in this area. This is partly due to the scarcity of publicly available standard educational material datasets for the evaluation of recommender systems. Unlike other domains, such as movies and books which have established public datasets for evaluation, public datasets for education domain recommenders are scarce. Hence the evaluation of ontology-based recommenders is difficult.

### **7.3 Future work**

Even if the overall results can be considered as positive, many aspects of this thesis can certainly be improved upon, and several promising future research directions may be outlined. In future, the repository will be enriched by absorbing more course and user information and heterogeneous data sources. In addition, it is planned to incorporate additional user contexts, e.g., student behaviour, learning styles and learning interests, into the recommendation process in order to make the system more comprehensive and intelligent. More feedback information from students may be employed, and the student model may be improved based on students' feedback, and consideration may be given to further aspects and techniques related to recommender systems. It is planned to carry out more experiments with a variety of actual students from different departments and from various academic backgrounds in order to prove the flexibility of this proposal.

In conclusion, this work has met both of its objectives, creating an OPCR and demonstrating the relevant recommender system. The OPCR offers three major components, data extraction, ontology mapping and a recommender system. The combination of these creates a novel framework within which a complicated problem can be solved. This work has made further contributions to the respective areas of its main components. The proof of concept system managed to complete a full operational cycle, from data gathering to course recommendations. Despite its infant state, there are clear indications that OPCR is able to reduce the phenomenon of information overloading.

## REFERENCES

- Abel, F., Bittencourt, I. I., Henze, N., Krause, D., & Vassileva, J. (2008). A rule-based recommender system for online discussion forums. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (Vol. 5149 LNCS, pp. 12–21). Springer. [https://doi.org/10.1007/978-3-540-70987-9\\_4](https://doi.org/10.1007/978-3-540-70987-9_4)
- Adibi, P., & Ladani, B. T. (2013). A collaborative filtering recommender system based on user's time pattern activity. In *The 5th Conference on Information and Knowledge Technology* (pp. 252–257). IEEE. <https://doi.org/10.1109/IKT.2013.6620074>
- Adomavicius, G., Kwon, Y., Tintarev, N., Masthoff, J., Ricci, F., Rokach, L., ... Kantor, P. B. (2011). *Recommender Systems Handbook*. *Recommender Systems Handbook* (Vol. 54). Springer. <https://doi.org/10.1007/978-0-387-85820-3>
- Adomavicius, G., & Tuzhilin, A. (2005a). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Adomavicius, G., & Tuzhilin, A. (2005b). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Agre, G., & Dongre, S. (2015). A Keyword Focused Web Crawler Using Domain Engineering and Ontology. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(3), 463–465. <https://doi.org/10.17148/IJARCCE.2015.43111>
- Agre, G. H., & Mahajan, N. V. (2015). Keyword focused web crawler. In *2nd International Conference on Electronics and Communication Systems, ICECS 2015* (pp. 1089–1092). IEEE. <https://doi.org/10.1109/ECS.2015.7124749>
- Al-Badarenah, A., & Alsakran, J. (2016). An Automated Recommender System for Course Selection. *International Journal of Advanced Computer Science and Applications*, 7(3), 166–175. <https://doi.org/10.14569/IJACSA.2016.070323>
- Alimam, M. A., & Seghiouer, H. (2013). Personalized E-Learning Environment based on Ontology to Improve Students ' School Levels. *International Journal of Computer Applications*, 84(12), 1–6.
- Almohammadi, K., Hagras, H., Alghazzawi, D., & Aldabbagh, G. (2017). A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms. *Journal of Artificial Intelligence and Soft Computing Research*, 7(1), 47–64. <https://doi.org/10.1515/jaiscr-2017-0004>
- Ambikapathy, R. (2011). Ontology Based E-Learning System for a University on Semantic Web. *International Journal of Mathematics and Computer Applications Research*, 1(2), 18–27.
- Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P., Gordijn, J., ... Weigand, H. (2006). Towards a Reference Ontology for Business Models (pp. 482–496). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/11901181\\_36](https://doi.org/10.1007/11901181_36)

- Antoniou, G., & Van Harmelen, F. (2008). A semantic Web primer. *Choice Reviews Online*, 46(03), 46-1523-46-1523. <https://doi.org/10.5860/CHOICE.46-1523>
- Apaza, R. G., Cervantes, E. V., Quispe, L. C., & Luna, J. O. (2014). Online courses recommendation based on LDA. In *CEUR Workshop Proceedings* (Vol. 1318, pp. 42–48).
- Bach, T.-L., & Dieng-Kuntz, R. (2005). Measuring similarity of elements in OWL DL ontologies. In *AAAI Workshop - Technical Report* (Vol. WS-05-01, pp. 96–99).
- Bagherifard, K., Rahmani, M., Nilashi, M., & Rafe, V. (2017). Performance improvement for recommender systems using ontology. *Telematics and Informatics*, 34, 1772–1792. <https://doi.org/10.1016/j.tele.2017.08.008>
- Bahramiana, Z., & Ali Abbaspoura, R. (2015). An ontology-based tourism recommender system based on Spreading Activation model. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(1W5), 83–90. <https://doi.org/10.5194/isprarchives-XL-1-W5-83-2015>
- Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66–72. <https://doi.org/10.1145/245108.245124>
- Beel, J., & Langer, S. (2015). A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In *International conference on theory and practice of digital libraries* (Vol. 9316, pp. 153–168). Springer. [https://doi.org/10.1007/978-3-319-24592-8\\_12](https://doi.org/10.1007/978-3-319-24592-8_12)
- Bendakir, N., & Aïmeur, E. (2006). Using Association Rules for Course Recommendation. In *AAAI Workshop on Educational Data Mining* (pp. 1–10).
- Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthaphaksin, R., & Nongpong, K. (2017). Recommender Systems for university elective course recommendation. *Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017*. <https://doi.org/10.1109/JCSSE.2017.8025933>
- Billsus, D., & Pazzani, M. J. (2000). User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction*, 10(2/3), 147–180. <https://doi.org/10.1023/A:1026501525781>
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>
- Bollen, D., Knijnenburg, B., Willemsen, M., & Graus, M. (2010). Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10* (p. 63). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1864708.1864724>
- Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. *Technology*, 243. <https://doi.org/10.1006/ijhc.1996.0096>
- Bozo, J., Alarcón, R., & Iribarra, S. (2010). Recommending learning objects according to a teachers' Context model. In *European Conference on Technology Enhanced Learning* (Vol. 6383 LNCS, pp. 470–475). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-16020-2\\_39](https://doi.org/10.1007/978-3-642-16020-2_39)

- Brace, I. (2005). *Questionnaire design: how to plan, structure and write survey material for effective market research*. *Choice Reviews Online* (Vol. 42). Kogan Page.  
<https://doi.org/10.5860/CHOICE.42-3520>
- Brown, C., Varley, P., & Pal, J. (2009). University course selection and services marketing. *Marketing Intelligence and Planning*, 27(3), 310–325.  
<https://doi.org/10.1108/02634500910955227>
- Bu, J., Tan, S., Chen, C., Wang, C., Wu, H., Zhang, L., & He, X. (2010). Music recommendation by unified hypergraph. In *Proceedings of the international conference on Multimedia - MM '10* (p. 391). New York, New York, USA: ACM Press.  
<https://doi.org/10.1145/1873951.1874005>
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4), 331–370.  
<https://doi.org/10.1023/A:1021240730564>
- Cantador, I., Bellogin, A., & Castells, P. (2008). Ontology-based personalised and context-aware recommendations of news items. In *2008 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* (pp. 562–565). IEEE.  
<https://doi.org/10.1109/WIAT.2008.204>
- Carballo, F. (2014). *Masters' Courses Recommendation : Exploring Collaborative Filtering and Singular Value Decomposition with Student Profiling*. MSc. Thesis. Instituto Superior Técnico, Lisbon, Portugal.
- Chang, P. C., Lin, C. H., & Chen, M. H. (2016). A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, 9(3).  
<https://doi.org/10.3390/a9030047>
- Chen, C.-M., Lee, H.-M., & Chen, Y.-H. (2005). Personalized e-learning system using Item Response Theory. *Computers & Education*, 44, 237–255.  
<https://doi.org/10.1016/j.compedu.2004.01.006>
- Claypool, M., Miranda, T., Gokhale, A., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Recommender Systems Workshop at ACM SIGIR* (pp. 40–48).  
<https://doi.org/10.1.1.46.3659>
- Coelho, B., Martins, C., & Almeida, A. (2010). Web intelligence in tourism: User modeling and recommender system. In *Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010* (Vol. 1, pp. 619–622).  
<https://doi.org/10.1109/WI-IAT.2010.236>
- Cone, D. C. (2011). *Introduction: Special Content Focus: International Emergency Medicine. Academic Emergency Medicine* (Vol. 18). <https://doi.org/10.1111/j.1553-2712.2011.01172.x>
- Cui, B., & Chen, X. (2009). An Online Book Recommendation System Based on Web Service. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery* (Vol. 7). <https://doi.org/10.1109/FSKD.2009.328>
- Daramola, O., Adigun, M., & Ayo, C. (2009). Building an Ontology-based Framework for Tourism Recommendation Services. In *Information and Communication Technologies in Tourism* (pp. 135–147). Springer.

- De Bruijn, J., Ehrig, M., Feier, C., Martíns-Recuerda, F., Scharffe, F., & Weiten, M. (2006). Ontology Mediation, Merging, and Aligning. In *Semantic Web Technologies: Trends and Research in Ontology-based Systems* (pp. 95–113). <https://doi.org/10.1002/047003033X.ch6>
- DeLong, C., Desikan, P., & Srivastava, J. (2006). USER: User-Sensitive Expert Recommendations for Knowledge-Dense Environments. In *Advances in Web Mining and Web Usage Analysis* (pp. 77–95). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/11891321\\_5](https://doi.org/10.1007/11891321_5)
- Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook* (pp. 107–144). Boston, MA: Springer US. [https://doi.org/10.1007/978-0-387-85820-3\\_4](https://doi.org/10.1007/978-0-387-85820-3_4)
- Diem, H. N. T. (2015). Personalized Course Recommendation in Formal Learning Based on Logistic Regression. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(10), 521–527. <https://doi.org/10.17148/IJARCCE.2015.410119>
- Drachsler, H., Hummel, H., & Koper, R. (2008a). Identifying the Goal, User model and Conditions of Recommender Systems for Formal and Informal Learning. Retrieved from <http://dspace.ou.nl/handle/1820/1187>
- Drachsler, H., Hummel, H., & Koper, R. (2008b). Navigation support for learners in informal learning environments. In *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08* (p. 303). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1454008.1454058>
- Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Van Rosmalen, P., ... Koper, R. (2009). ReMashed - Recommendation approaches for mash-up personal learning environments in formal and informal learning settings. In *CEUR Workshop Proceedings* (Vol. 506, pp. 23–30). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-04636-0\\_85](https://doi.org/10.1007/978-3-642-04636-0_85)
- DS, G., & K, K. (2015). Developing an Intelligent Recommendation System for Course Selection by Students for Graduate Courses. *Business and Economics Journal*, 7(2). <https://doi.org/10.4172/2151-6219.1000209>
- Ekstrand, M. D., Harper, F. M., Willemsen, M. C., & Konstan, J. A. (2014). User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14* (pp. 161–168). <https://doi.org/10.1145/2645710.2645737>
- Farzan, R., & Brusilovsky, P. (2006). Social navigation support in a course recommendation system. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 4018 LNCS, pp. 91–100). [https://doi.org/10.1007/11768012\\_11](https://doi.org/10.1007/11768012_11)
- Fernandez-Lopez, M., & Corcho, O. (2010). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Retrieved from <https://dl.acm.org/citation.cfm?id=1965527>
- Gabrilovich, E., Gabrilovich, E., Markovitch, S., & Markovitch, S. (2006). Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. *Proceedings of the Twenty-First National Conference on*

- Artificial Intelligence* (Vol. telligence). <https://doi.org/10.1.1.66.3456>
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). Sweetening Ontologies with DOLCE (pp. 166–181). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-45810-7\\_18](https://doi.org/10.1007/3-540-45810-7_18)
- Garcia-Martinez, S., & Hamou-Lhdj, A. (2013). Educational Recommender Systems: A Pedagogical-Focused Perspective. In *Smart Innovation, Systems and Technologies* (Vol. 25, pp. 113–124). Springer, Heidelberg. [https://doi.org/10.1007/978-3-319-00375-7\\_8](https://doi.org/10.1007/978-3-319-00375-7_8)
- Garcia, I., Sebastia, L., & Onaindia, E. (2011). On the design of individual and group recommender systems for tourism. *Expert Systems With Applications*, 38, 7683–7692. <https://doi.org/10.1016/j.eswa.2010.12.143>
- Gasparini, I., Lichtnow, D., Pimenta, M. S., & De Oliveira, J. P. M. (2009). Quality ontology for recommendation in an adaptive educational system. *International Conference on Intelligent Networking and Collaborative Systems, INCOS 2009*, 329–334. <https://doi.org/10.1109/INCOS.2009.11>
- Gasparini, I., Lichtnow, D., Pimenta, M. S., & Oliveira, J. P. M. de. (2009). Quality Ontology for Recommendation in an Adaptive Educational System. In *2009 International Conference on Intelligent Networking and Collaborative Systems* (pp. 329–334). IEEE. <https://doi.org/10.1109/INCOS.2009.11>
- Gautam Pant, & Padmini Srinivasan. (2006). Link contexts in classifier-guided topical crawlers. *IEEE Transactions on Knowledge and Data Engineering*, 18(1), 107–122. <https://doi.org/10.1109/TKDE.2006.12>
- Ge, J., Chen, Z., Peng, J., & Li, T. (2012). An ontology-based method for personalized recommendation. *Proceedings of the 11th IEEE International Conference on Cognitive Informatics and Cognitive Computing, ICCI\*CC 2012*, 522–526. <https://doi.org/10.1109/ICCI-CC.2012.6311204>
- Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004). S-MATCH: Algorithm and implementation. In *European semantic web symposium*, pp. 61-75. Springer (Vol. 3053, pp. 61–75). Berlin, Heidelberg.
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61–70. <https://doi.org/10.1145/138859.138867>
- Grazia, M., Bono, D., Pieri, G., & Salvetti, O. (2004). WP9: A Review of Data and Metadata Standards and Techniques for Representation of Multimedia Content. *MUSCLE Project Deliverable, IST-CNR*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.668.47&rep=rep1&type=pdf>
- Group, D. E. (2009). *Leveraging Content from Open Corpus Sources for Technology Enhanced Learning. October*. [https://doi.org/10.1109/10.1007/978-3-642-02451-1\\_27](https://doi.org/10.1109/10.1007/978-3-642-02451-1_27)
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220. <https://doi.org/10.1006/knac.1993.1008>
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. San Francisco, CA, itd: Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-381479-1.00001-0>

- Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00* (pp. 241–250). New York, New York, USA: ACM Press. <https://doi.org/10.1145/358916.358995>
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53. <https://doi.org/10.1145/963770.963772>
- Hernández del Olmo, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 35, 790–804. <https://doi.org/10.1016/j.eswa.2007.07.047>
- Hongji Yang, Zhan Cui, & O'Brien, P. (1999). Extracting ontologies from legacy systems for understanding and re-engineering. *Proceedings. Twenty-Third Annual International Computer Software and Applications Conference (Cat. No.99CB37032)*, 21–26. <https://doi.org/10.1109/CMP SAC.1999.812512>
- Hsu, M. H. (2008). A personalized English learning recommender system for ESL students. *Expert Systems with Applications*, 34(1), 683–688. <https://doi.org/10.1016/j.eswa.2006.10.004>
- Hu, J., & Zhang, W. (2008). Community Collaborative Filtering for E-Learning. In *2008 International Conference on Computer and Electrical Engineering* (pp. 593–597). IEEE. <https://doi.org/10.1109/ICCEE.2008.144>
- Hu, R. (2010). Design and user issues in personality-based recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10* (p. 357). <https://doi.org/10.1145/1864708.1864790>
- Huang, C. Y., Chen, R. C., & Chen, L. S. (2013). Course-recommendation system based on ontology. *Proceedings - International Conference on Machine Learning and Cybernetics*, 3, 1168–1173. <https://doi.org/10.1109/ICMLC.2013.6890767>
- Huang, T., Zhan, G., Zhang, H., & Yang, H. (2017). MC RS: A course recommendation system for MOOCs. In *Multimedia Tools and Applications* (Vol. 68, pp. 1–19). <https://doi.org/10.1007/s11042-017-4620-2>
- Huang, Y. (2017). *A Hybrid Recommender: Study and implementation of course selection recommender engine*. MSc. Thesis. Uppsala Univ., Uppsala, Sweden. Retrieved from <http://www.teknat.uu.se/student%0Ahttps://uu.diva-portal.org/smash/get/diva2:1092321/FULLTEXT01.pdf>
- Ibrahim, M. E., Yang, Y., & Ndzi, D. (2017). Using ontology for personalised course recommendation applications. In *International Conference on Computational Science and Its Application* (Vol. 10404, pp. 426–438). Springer, Cham. [https://doi.org/10.1007/978-3-319-62392-4\\_31](https://doi.org/10.1007/978-3-319-62392-4_31)
- Ibrahim, M. E., Yang, Y., Ndzi, D., Yang, G., & Almaliki, M. (2018). Ontology-based Personalised Course Recommendation Framework. *IEEE Access*, 1–1. <https://doi.org/10.1109/ACCESS.2018.2889635>
- Jambhulkar, S. V., & Karale, S. J. (2017). Semantic web application generation using Protégé tool. In *Proceedings of 2016 Online International Conference on Green Engineering and Technologies, IC-GET 2016* (pp. 1–5). <https://doi.org/10.1109/GET.2016.7916686>

- Jannach, D., Zanker, M., Felfering, A., & Friedrich, G. (2011). Recommender Systems An Introduction. *Teaching RS*, 1–335. <https://doi.org/10.1017/CBO9780511763113>
- Jones, N., & Pu, P. (2009). *User Acceptance Issues In Music Recommender Systems. EPFL Technical Report HCI-REPORT-2009-001*. Retrieved from [https://infoscience.epfl.ch/record/142722/files/tr\\_accept\\_njones.pdf](https://infoscience.epfl.ch/record/142722/files/tr_accept_njones.pdf)
- Kadima, H., & Malek, M. (2010). Toward ontology-based personalization of a recommender system in social network. *Proceedings of the 2010 International Conference of Soft Computing and Pattern Recognition, SoCPaR 2010*, 5, 119–122. <https://doi.org/10.1109/SOCPAR.2010.5685957>
- Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1), S0269888903000651. <https://doi.org/10.1017/S0269888903000651>
- Kalyuga, S. (2011). Effects of information transiency in multimedia learning. *Procedia - Social and Behavioral Sciences*, 30, 307–311. <https://doi.org/10.1016/j.sbspro.2011.10.061>
- Kaminskas, M., & Bridge, D. (2014). A comprehensive survey of neighborhood-based recommendation methods. In *RecSys REDD 2014: International Workshop on Recommender Systems Evaluation: Dimensions and Design* (Vol. 69, pp. 2–7). [https://doi.org/10.1007/978-0-387-85820-3\\_4](https://doi.org/10.1007/978-0-387-85820-3_4)
- Kawtrakul, A., Suktarachan, M., & Aurawan Imsombut. (2003). Automatic Thai Ontology Construction and Maintenance System. Retrieved from <http://www.fao.org/agrovoc>
- Kembellec, G., Chartron, G., & Saleh, I. (2014). Recommender systems. *Recommender Systems*, 40(3), 1–232. <https://doi.org/10.1002/9781119054252>
- Khan, L. R. (2000). *Ontology-based Information Selection*. Ph.D. Thesis. South California Univ. ,Los Angeles, California. [https://doi.org/10.1016/S0040-6090\(00\)01805-8](https://doi.org/10.1016/S0040-6090(00)01805-8)
- Kim, Y. S. (2013). Recommender system based on product taxonomy in E-commerce sites. *Journal of Information Science and Engineering*, 29, 63–78.
- Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In *The Semantic Web--ISWC 2004* (pp. 229–243). <https://doi.org/10.1.1.89.9268>
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37. <https://doi.org/10.1109/MC.2009.263>
- Kumar, N. (2011). A computational study of metabolism in sprinting. In *International Journal of Engineering, Transactions B: Applications* (Vol. 24, pp. 75–80). New York, New York, USA: ACM Press. <https://doi.org/10.1145/371920.372071>
- Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *Proceedings of the 12th International Machine Learning Conference (ML95)*, 331–339. <https://doi.org/10.1.1.22.6286>
- Lee, Y. (2011). An Intelligent Course Recommendation System. *The Smart Computing Review*, 1(1), 69–84. <https://doi.org/10.6029/smarterc.2011.01.006>
- Li, T., Ogihara, M., & Li, Q. (2003). A comparative study on content-based music genre

- classification. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '03* (p. 282). New York, New York, USA: ACM Press. <https://doi.org/10.1145/860435.860487>
- Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(February), 76–80. <https://doi.org/10.1109/MIC.2003.1167344>
- Liu, J., Wang, X., Liu, X., & Yang, F. (2010). Analysis and design of personalized recommendation system for university physical education. *2010 International Conference on Networking and Digital Society, ICNDS 2010*, 2, 472–475. <https://doi.org/10.1109/ICNDS.2010.5479470>
- Lops, P., de Gemmis, M., & Semeraro, G. (2011). *Content-based Recommender Systems: State of the Art and Trends. Recommender Systems Handbook*. [https://doi.org/10.1007/978-0-387-85820-3\\_3](https://doi.org/10.1007/978-0-387-85820-3_3)
- Lotfy, M. M., & Salama, a a. (2014). Subject Recommendation Using Ontology for Computer Science ACM Curricula. *International Journal of Information Science and Intelligent System*, 3(1), 199–205.
- Maes, P., & Sharadanand, M. S. (1995). *Social information filtering: algorithms for automating "Word of Mouth"*, CHI Proc. <https://doi.org/10.1145/223904.223931>
- Maidel, V., Shoval, P., Shapira, B., & Taieb-Maimon, M. (2010). Ontological content-based filtering for personalised newspapers. *Online Information Review*, 34(5), 729–756. <https://doi.org/10.1108/14684521011084591>
- Manouselis, N., & Costopoulou, C. (2006). Designing a Web-based Testing Tool for Multi-Criteria Recommender Systems. In *Engineering Letters* (Vol. 13, pp. 417–427). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.3387&rep=rep1&type=pdf>
- Manouselis, N., Drachsler, H., Verbert, K., Santos, O. C., & Konstan, J. A. (2014). Recommender systems for technology enhanced learning: Research trends and applications. In N. Manouselis, H. Drachsler, K. Verbert, & O. C. Santos (Eds.), *International Conference on Web-Based Learning* (pp. 1–306). New York, NY: Springer New York. <https://doi.org/10.1007/978-1-4939-0530-0>
- Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., & Koper, R. (2011). Recommender Systems in Technology Enhanced Learning. In *Recommender Systems Handbook* (pp. 387–415). [https://doi.org/10.1007/978-0-387-85820-3\\_12](https://doi.org/10.1007/978-0-387-85820-3_12)
- Manouselis, N., & Vuorikari, R. (2009). What if annotations were reusable: A preliminary discussion. In *International Conference on Web-Based Learning* (Vol. 5686 LNCS, pp. 255–264). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-03426-8\\_32](https://doi.org/10.1007/978-3-642-03426-8_32)
- Manouselis, N., Vuorikari, R., & Van Assche, F. (2010). Collaborative recommendation of e-learning resources: An experimental investigation. *Journal of Computer Assisted Learning*, 26(4), 227–242. <https://doi.org/10.1111/j.1365-2729.2010.00362.x>
- Masters, J., Madhyastha, T., & Shakouri, A. (2008). ExplaNet: A collaborative learning tool and hybrid recommender system for student-authored explanations. *Journal of Interactive Learning Research*, 19(1), 51–74.

- Middleton, S. E., De Roure, D., & Shadbolt, N. R. (2009). Ontology-Based Recommender Systems. In *Handbook on Ontologies* (pp. 779–796). Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-92673-3>
- Mooney, R. J., & Roy, L. (1999). Content-Based Book Recommending Using Learning for Text Categorization. *Proceedings of the Fifth ACM Conference on Digital Libraries - DL '00*, 195–204. <https://doi.org/10.1145/336597.336662>
- Murthy, S., Kasif, S., Salzberg, S., & Beigel, R. (1993). *OC1: A randomized algorithm for building oblique decision trees*. Proc. 11th American Association for Artificial Intelligence Conference {(AAAI'93)}. <https://doi.org/10.1002/cjoc.201500640>
- Musto, C. (2010). Enhanced vector space models for content-based recommender systems. *Proceedings of the Fourth ACM Conference on Recommender Systems*, 361–364. <https://doi.org/10.1145/1864708.1864791>
- Noakes, J. E., Arrott, A., & Haakana, C. (1968). Vibrating sample magnetometers. In *Review of Scientific Instruments* (Vol. 39, pp. 1436–1438). <https://doi.org/10.1063/1.1683127>
- Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. W., & Musen, M. A. (2001). Creating Semantic Web Contents with Protégé-2000. *Intelligent Systems, IEEE*, 16(2), 60–71.
- O'Mahony, M. P., & Smyth, B. (2007). A recommender system for on-line course enrolment: an initial study. In *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07* (p. 133). <https://doi.org/10.1145/1297231.1297254>
- Obeid, C., Lahoud, I., El Khoury, H., & Champin, P.-A. (2018). Ontology-based Recommender System in Higher Education. In *Companion of the The Web Conference 2018 on The Web Conference 2018* (Vol. 2, pp. 1031–1034). ACM. <https://doi.org/10.1145/3184558.3191533>
- Omelayenko, B. (2002). Integrating Vocabularies: Discovering and Representing Vocabulary Maps. In *First International Semantic Web Workshop* (Vol. 2342, pp. 206–220). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-48005-6\\_17](https://doi.org/10.1007/3-540-48005-6_17)
- Ostuni, V. C., Di Noia, T., Mirizzi, R., & Di Sciascio, E. (2014). Top-N recommendations from implicit feedback leveraging linked open data. In *CEUR Workshop Proceedings* (Vol. 1127, pp. 20–27). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2507157.2507172>
- Pant, G., & Menczer, F. (2003). Topical Crawling for Business Intelligence. In *7th European Conference on Research and Advanced Technology for Digital Libraries ECDL 2003* (Vol. 2769, pp. 233–244). Springer, Berlin, Heidelberg. <https://doi.org/10.1007/s00296-007-0405-7>
- Pant, G., Srinivasan, P., & Menczer, F. (2004). Crawling the Web. In *Web Dynamics* (pp. 153–177). [https://doi.org/10.1007/978-3-662-10874-1\\_7](https://doi.org/10.1007/978-3-662-10874-1_7)
- Parameswaran, A., Venetis, P., & Garcia-Molina, H. (2011). Recommendation Systems with Complex Constraints: A Course Recommendation Perspective. *ACM Trans. Inf. Syst.*, 29, 33. <https://doi.org/10.1145/2037661.2037665>
- Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012, September 1). A literature review and classification of recommender systems research. *Expert Systems with Applications*. Pergamon. <https://doi.org/10.1016/j.eswa.2012.02.038>

- Park, Y. (2017). Recommending Personalized Tips on New Courses for Guiding Course Selection. *Proceedings of the SouthEast Conference on - ACM SE '17*, 172–174. <https://doi.org/10.1145/3077286.3077567>
- Pazzani, M., & Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27(3), 313–331. <https://doi.org/10.1023/A:1007369909943>
- Pazzani, M. J. (1999). *A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review* (Vol. 13). <https://doi.org/10.1023/A:1006544522159>
- Pazzani, M. J., & Billsus, D. (2007a). Content-Based Recommendation Systems. In *The Adaptive Web* (pp. 325–341). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-72079-9\\_10](https://doi.org/10.1007/978-3-540-72079-9_10)
- Pazzani, M. J., & Billsus, D. (2007b). Content-Based Recommendation Systems. In *The Adaptive Web* (pp. 325–341). [https://doi.org/10.1007/978-3-540-72079-9\\_10](https://doi.org/10.1007/978-3-540-72079-9_10)
- Pearce, J. (2008). *A System to Encourage Playful Exploration in a Reflective Environment. ED-MEDIA 2008--World Conference on Educational Multimedia, Hypermedia & Telecommunications.*
- Prieto, M. E., Menéndez, V. H., Segura, A. A., & Vidal, C. L. (2008). A Recommender System Architecture for Instructional Engineering. In *Emerging Technologies and Information Systems for the Knowledge Society* (pp. 314–321). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-87781-3\\_35](https://doi.org/10.1007/978-3-540-87781-3_35)
- Pu, P., & Chen, L. (2010). A User-Centric Evaluation Framework of Recommender Systems. In *ACM Transactions on Computer-Human Interaction* (Vol. 17, pp. 1–33). <https://doi.org/10.1145/1721831.1721836>
- Punj, G. N., & Moore, R. (2007). Smart versus knowledgeable online recommendation agents. *Journal of Interactive Marketing*, 21(4), 46–60. <https://doi.org/10.1002/dir.20089>
- Rafaeli, S., Dan-Gur, Y., & Barak, M. (2005). Social Recommender Systems. *International Journal of Distance Education Technologies*, 3(2), 30–47. <https://doi.org/10.4018/jdet.2005040103>
- Ramadoss, B., & Balasundaram, S. R. (2006). Management and selection of visual metaphors for courseware development in web based learning. In *2006 IEEE Conference on Cybernetics and Intelligent Systems* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICCIS.2006.252345>
- Ray, S., & Sharma, A. (2011). *A collaborative filtering based approach for recommending elective courses. Communications in Computer and Information Science* (Vol. 141 CCIS). [https://doi.org/10.1007/978-3-642-19423-8\\_34](https://doi.org/10.1007/978-3-642-19423-8_34)
- Ren, R., Zhang, L., Cui, L., Deng, B., & Shi, Y. (2015). Personalized financial news recommendation algorithm based on ontology. In *Procedia Computer Science* (Vol. 55, pp. 843–851). <https://doi.org/10.1016/j.procs.2015.07.151>
- Ringe, S., Francis, N., & A, P. A. H. S. (2012). Ontology Based Web Crawler. *International Journal of Computer Applications in Engineering Sciences*, II(194), 194–198.
- Rinku, M. A. R., & Aravind, A. (2016). Children's mind: Cultural values in Chhota Bheem.

- Ruotsalo, T., & Hyvönen, E. (2007). A Method for Determining Ontology-Based Semantic Relevance. *18th International Conference, DEXA 2007*, 4653, 680–688. [https://doi.org/10.1007/978-3-540-74469-6\\_66](https://doi.org/10.1007/978-3-540-74469-6_66)
- Ruthven, I., & Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(2). <https://doi.org/10.1017/S0269888903000638>
- Salahli, M. A., Özdemir, M., & Yaşar, C. (2013). Concept based approach for adaptive personalized course learning system. *International Education Studies*, 6(5), 92–103. <https://doi.org/10.5539/ies.v6n5p92>
- Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *24th international conference on Machine learning* (pp. 791–798). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1273496.1273596>
- Sandvig, J., & Burke, R. (2005). *Aacorn: A CBR recommender for academic advising.* Technical Report TR05-015. School of Computer Science, Telecommunications, and Information Systems , DePaul Univ. Chicago, USA. Retrieved from <http://facweb.cti.depaul.edu/research/TechReports/TR05-015.doc>
- Santos, O. C., & Boticario, J. G. (2009). Guiding learners in learning management systems through recommendations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5794 LNCS, pp. 596–601). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-04636-0\\_54](https://doi.org/10.1007/978-3-642-04636-0_54)
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce - EC '00* (pp. 158–167). New York, New York, USA: ACM Press. <https://doi.org/10.1145/352871.352887>
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *the tenth international conference on World Wide Web - WWW '01* (Vol. 1, pp. 285–295). ACM Press. <https://doi.org/10.1145/371920.372071>
- Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02* (p. 253). <https://doi.org/10.1145/564376.564421>
- Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. In *Recommender Systems Handbook* (pp. 257–297). [https://doi.org/10.1007/978-0-387-85820-3\\_8](https://doi.org/10.1007/978-0-387-85820-3_8)
- Shrivastav, H., & Hiltz, S. R. (2013). Information overload in technology-based education: A meta-analysis. In *19th Americas Conference on Information Systems*.
- Shvaiko, P., & Euzenat, J. (2013). Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(X), 158–176. <https://doi.org/10.1109/TKDE.2011.253>

- Sieg, A., Mobasher, B., & Burke, R. (2010). Ontology-based collaborative recommendation. *CEUR Workshop Proceedings*, 606, 20–31.
- Singto, P., & Mingkhwan, A. (2013). Semantic Searching IT Careers Concepts Based on Ontology. *Journal of Advanced Management Science*, 1(1), 102–106. <https://doi.org/10.12720/joams.1.1.102-106>
- Sinha, R., & Swearingen, K. (2002). The role of transparency in recommender systems. In *CHI '02 extended abstracts on Human factors in computing systems - CHI '02* (p. 830). <https://doi.org/10.1145/506443.506619>
- Smyth, B., & Cotter, P. (2000). Personalized TV listings service for the digital TV age. *Knowledge-Based Systems*, 13(2), 53–59. [https://doi.org/10.1016/S0950-7051\(00\)00046-0](https://doi.org/10.1016/S0950-7051(00)00046-0)
- Tang, T., & McCalla, G. (2004). Laws of Attraction: In Search of Document Value-ness for Recommendation. In *Research and Advanced Technology for Digital Libraries* (pp. 269–280). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-30230-8\\_25](https://doi.org/10.1007/978-3-540-30230-8_25)
- Tang, T. Y., & McCalla, G. (2009a). A multidimensional paper recommender: Experiments and evaluations. *IEEE Internet Computing*, 13(4), 34–41. <https://doi.org/10.1109/MIC.2009.73>
- Tang, T. Y., & McCalla, G. (2009b). A Multidimensional Paper Recommender: Experiments and Evaluations. *IEEE Internet Computing*, 13(4), 34–41. <https://doi.org/10.1109/MIC.2009.73>
- ucas.com. (2018). The Universities and Colleges Admissions Service in United Kingdom. Retrieved October 15, 2018, from <https://www.ucas.com/>
- Vallet, D., Castells, P., Fernández, M., Mylonas, P., & Avrithis, Y. (2007). Personalized content retrieval in context using ontological knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3), 336–345. <https://doi.org/10.1109/TCSVT.2007.890633>
- Wan, X., Ninomiya, T., & Okamoto, T. (2008). LRMDCR: A Learner's Role-Based Multi Dimensional Collaborative Recommendation for Group Learning Support. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies* (pp. 603–605). IEEE. <https://doi.org/10.1109/ICALT.2008.63>
- Wang, F.-H. (2008). Content Recommendation Based on Education-Contextualized Browsing Events for Web-based Personalized Learning. *Journal of Educational Technology & Society*, 11(4), 94–112.
- Wang, S., & Sapporo, K. (2006). Topic-Oriented Query Expansion for Web Search Categories and Subject Descriptors. In *Proceedings of the 15th international conference on World Wide Web* (pp. 1029–1030).
- Yang, J. C., Huang, Y. T., Tsai, C. C., Chung, C. I., & Wu, Y. C. (2009). An Automatic Multimedia Content Summarization System for Video Recommendation. *Educational Technology & Society*, 12(1), 49–61.
- Yang, Q., Sun, J., Wang, J., & Jin, Z. (2010). Semantic web-based personalized recommendation system of courses knowledge research. In *International Conference on Intelligent Computing and Cognitive Informatics, ICICCI 2010* (pp. 214–217). IEEE. <https://doi.org/10.1109/ICICCI.2010.54>

- Yang, S. Y. (2010). Developing an ontology-supported information integration and recommendation system for scholars. *Expert Systems with Applications*, 37(10), 7065–7079. <https://doi.org/10.1016/j.eswa.2010.03.011>
- Yang, Y. J., & Wu, C. (2009). An attribute-based ant colony system for adaptive learning object recommendation. *Expert Systems with Applications*, 36(2 PART 2), 3034–3047. <https://doi.org/10.1016/j.eswa.2008.01.066>
- Zemmouch, L., & Ghomari, A. R. (2013). Process of Building Reference Ontology for Higher Education. In *World Congress on Engineering - Wce 2013, Vol Iii* (Vol. III, pp. 1595–1600).
- Zeng, L., Zhu, T., & Ding, X. (2009). Study on construction of university course ontology: Content, method and process. In *International Conference on Computational Intelligence and Software Engineering, CiSE 2009* (pp. 1–4). IEEE. <https://doi.org/10.1109/CISE.2009.5363158>
- Zezula, P., Dohnal, V., & Amato, G. (2006). *Similarity Search - The Metric Space Approach. Science & Business Media*. Springer Science & Business Media.
- Zhang, H., Yang, H., Huang, T., & Zhan, G. (2017). DBNCF: Personalized courses recommendation system based on DBN in MOOC environment. *Proceedings - 2017 International Symposium on Educational Technology, ISET 2017*, 106–108. <https://doi.org/10.1109/ISET.2017.33>
- Zhang, L. (2013). The definition of novelty in recommendation system. *Journal of Engineering Science and Technology Review*, 6(3), 141–145. <https://doi.org/10.7860/JCDR/2013/6448.3878>
- Zhang, L., & Li, J. (2011). Automatic Generation of Ontology Based on Database Trajectory analysis View project Automatic Generation of Ontology Based on Database. *Journal of Computational Information Systems*, 7, 1148–1154.
- Zhang, X. (2009). Civil engineering professional courses collaborative recommendation system based on network. *2009 1st International Conference on Information Science and Engineering, ICISE 2009*, 3253–3256. <https://doi.org/10.1109/ICISE.2009.392>
- Zhang, Y., Callan, J., & Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02* (p. 81). New York, New York, USA: ACM Press. <https://doi.org/10.1145/564376.564393>
- Zhang, Z., Zhou, T., & Zhang, C. (2010). Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *Physica A: Statistical Mechanics and Its Applications*, 389(1), 179–186. <https://doi.org/10.1016/j.physa.2009.08.036>
- Zhao, Z. D., & Shang, M. S. (2010). User-based collaborative-filtering recommendation algorithms on hadoop. In *3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010* (pp. 478–481). IEEE. <https://doi.org/10.1109/WKDD.2010.54>
- Zhou, K., Yang, S.-H., & Zha, H. (2011). *Functional matrix factorizations for cold-start recommendation*. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*. <https://doi.org/10.1145/2009916.2009961>

Zhou, X., Wu, Z., Yin, A., Wu, L., Fan, W., & Zhang, R. (2004). *Ontology development for unified traditional Chinese medical language system*. *Artificial Intelligence in Medicine* (Vol. 32). Stanford Univ. Stanford, California.  
<https://doi.org/10.1016/j.artmed.2004.01.014>

## APPENDICES

### Appendix A Course subject classifications

The courses subjects were classified based on HECoS (The Higher Education Classification of Subjects). Please refer to Appendix A and tables A.1 and A.2 which show the classification of the subjects in the fields of information technology and business and administration.

Field of Study	Main Subject	Major subject
Computer science	Computer science	Computer architectures & operating systems
Computer science	Computer science	Computer architectures
Computer science	Computer science	Operating systems
Computer science	Computer science	Displays & imaging
Computer science	Computer science	High end computing
Computer science	Computer science	Parallel computing
Computer science	Computer science	Networks & communications
Computer science	Computer science	Computational science foundations
Computer science	Computer science	Human-computer interaction
Computer science	Computer science	Multimedia computing science
Computer science	Computer science	Internet
Computer science	Computer science	e-business
Computer science	Information systems	Systems design methodologies
Computer science	Information systems	Systems analysis & design
Computer science	Information systems	Databases
Computer science	Information systems	Data management
Computer science	Information systems	Intelligent & expert systems
Computer science	Software engineering	Software engineering
Computer science	Software engineering	Software design
Computer science	Software engineering	Programming
Computer science	Software engineering	Procedural programming
Computer science	Software engineering	Object-oriented programming
Computer science	Software engineering	Declarative programming
Computer science	Artificial intelligence	Speech & natural language processing
Computer science	Artificial intelligence	Knowledge representation
Computer science	Artificial intelligence	Neural computing
Computer science	Artificial intelligence	Computer vision
Computer science	Artificial intelligence	Cognitive modelling
Computer science	Artificial intelligence	Machine learning
Computer science	Artificial intelligence	Automated reasoning
Computer science	Health informatics	Health technologies
Computer science	Health informatics	Bioinformatics
Computer science	Health informatics	Tele healthcare
Computer science	Games	Computer games programming
Computer science	Games	Computer games design
Computer science	Games	Computer games graphics
Computer science	Computer generated visual & audio effects	Computer generated visual & audio effects
Computer science	Computer generated visual & audio effects	Computer generated imagery

**Table A.1** course subject classification in Computer science

<b>Field of Study</b>	<b>Main Subject</b>	<b>Major subject</b>
Business & administrative studies	Business studies	European business studies
Business & administrative studies	Business studies	International business studies
Business & administrative studies	Management studies	Management studies
Business & administrative studies	Management studies	Management techniques
Business & administrative studies	Management studies	Strategic management
Business & administrative studies	Management studies	Creative management
Business & administrative studies	Management studies	Project management
Business & administrative studies	Management studies	Change management
Business & administrative studies	Management studies	Organisational development
Business & administrative studies	Management studies	Institutional management
Business & administrative studies	Management studies	Domestic management
Business & administrative studies	Management studies	Criminal justice management
Business & administrative studies	Management studies	Land & property management
Business & administrative studies	Management studies	Land management
Business & administrative studies	Management studies	Property management
Business & administrative studies	Management studies	Property valuation & auctioneering
Business & administrative studies	Management studies	Retail management
Business & administrative studies	Management studies	Emergency & disaster management
Business & administrative studies	Management studies	Emergency services management
Business & administrative studies	Management studies	Disaster management
Business & administrative studies	Finance	Banking
Business & administrative studies	Finance	Investment & insurance
Business & administrative studies	Finance	Investment
Business & administrative studies	Finance	Insurance
Business & administrative studies	Finance	Actuarial science
Business & administrative studies	Finance	Taxation
Business & administrative studies	Finance	Financial management
Business & administrative studies	Finance	Financial risk
Business & administrative studies	Accounting	Accountancy
Business & administrative studies	Accounting	Cost & management accountancy
Business & administrative studies	Accounting	Public accountancy
Business & administrative studies	Accounting	Book keeping
Business & administrative studies	Accounting	Accounting theory
Business & administrative studies	Accounting	Auditing of accounts
Business & administrative studies	Accounting	Financial reporting
Business & administrative studies	Marketing	Market research
Business & administrative studies	Marketing	Sales management
Business & administrative studies	Marketing	Distribution
Business & administrative studies	Marketing	International marketing
Business & administrative studies	Marketing	Promotion & advertising
Business & administrative studies	Marketing	Advertising
Business & administrative studies	Marketing	Corporate image
Business & administrative studies	Marketing	Sponsorship
Business & administrative studies	Human resource management	Industrial relations
Business & administrative studies	Human resource management	Staff development
Business & administrative studies	Human resource management	Training methods
Business & administrative studies	Human resource management	Recruitment methods
Business & administrative studies	Human resource management	(N620) Health & safety issues
Business & administrative studies	Office skills	Office administration
Business & administrative studies	Office skills	Secretarial & typing skills
Business & administrative studies	Office skills	Audio typing
Business & administrative studies	Office skills	Shorthand & shorthand transcription

**Table A.2** course subject classification in business and management

## Appendix B Similarity calculation function process in CBF and CF

Functions process to calculate feature similarity in CBF part in OPRCourse. Describe implementation functions process in CBF and how generate course vector and user vector, similarity between course profile features such as course title, course fee, location:

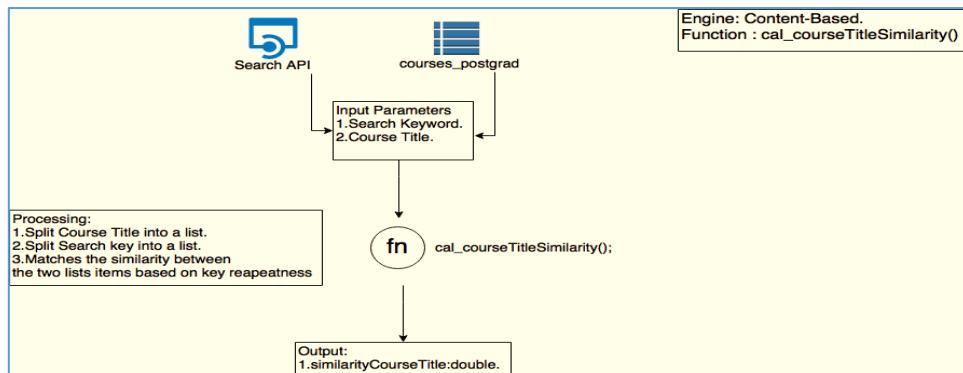


Figure B.1 Course title similarity calculation function process

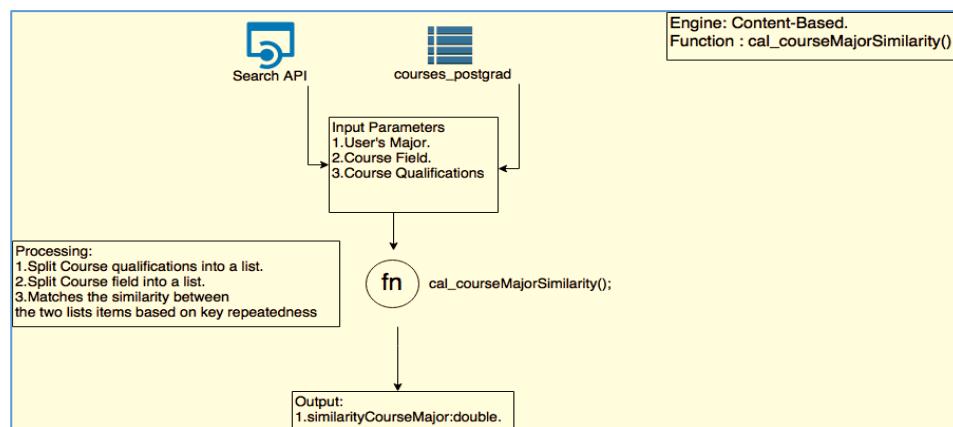


Figure B.2 Course major subject similarity calculation function process

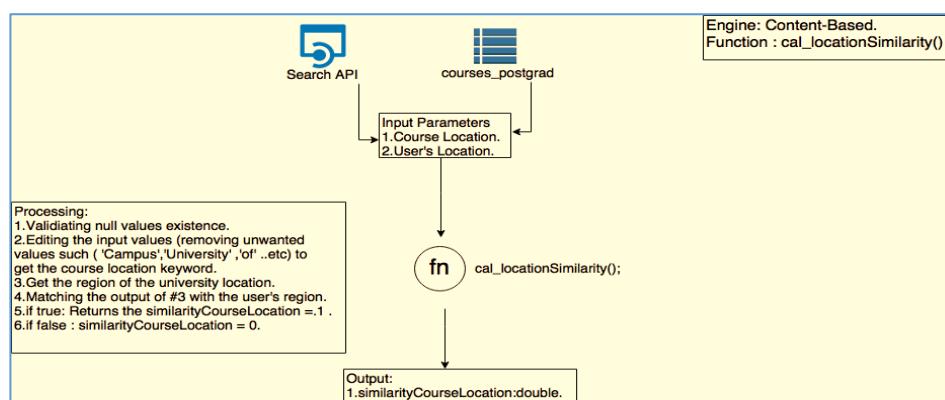


Figure B.3 University location similarity calculation function process

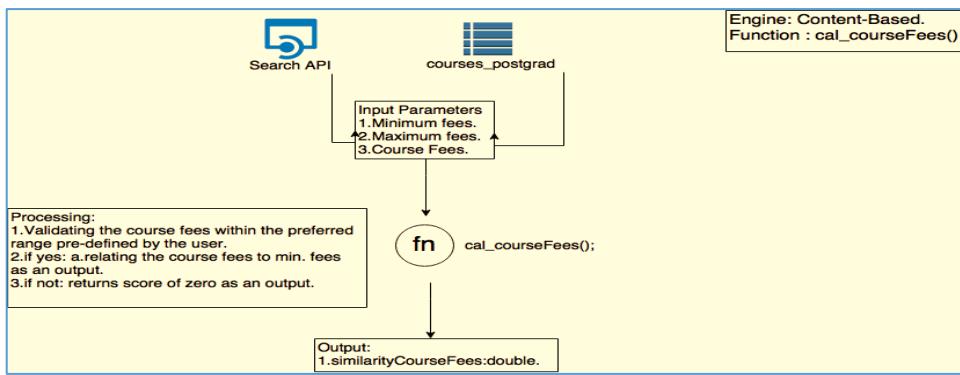


Figure B.4 Course fee similarity calculation function process

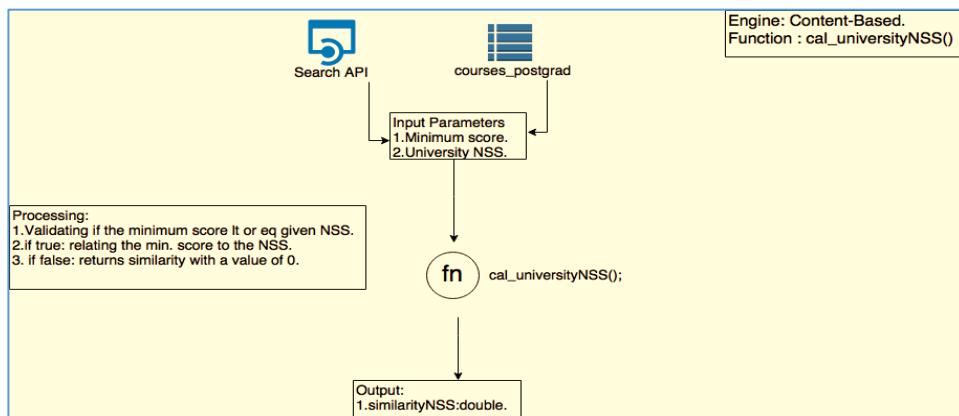


Figure B.5 University NSS score similarity calculation function process

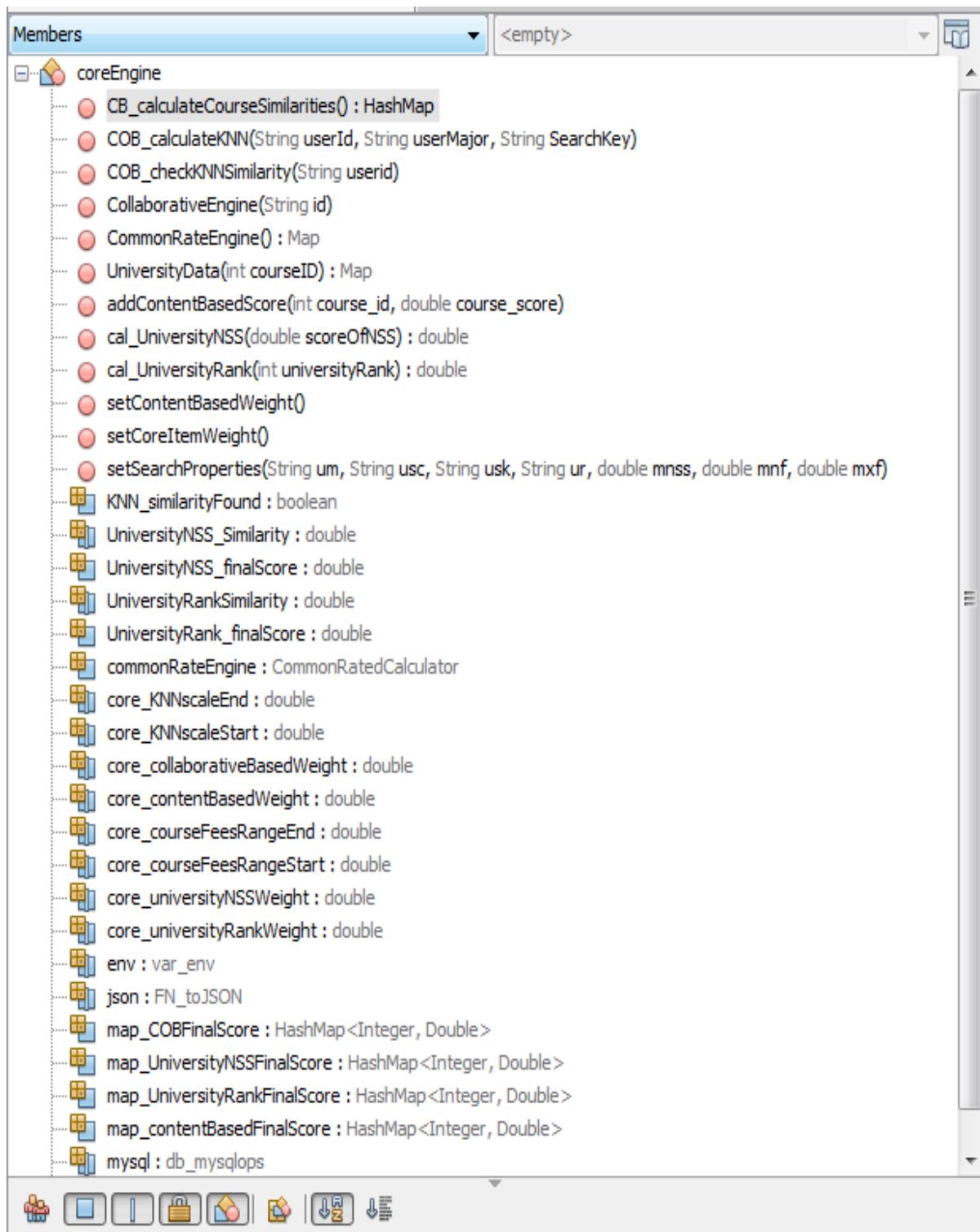


Figure B. 6 CBF functions and classes in the code

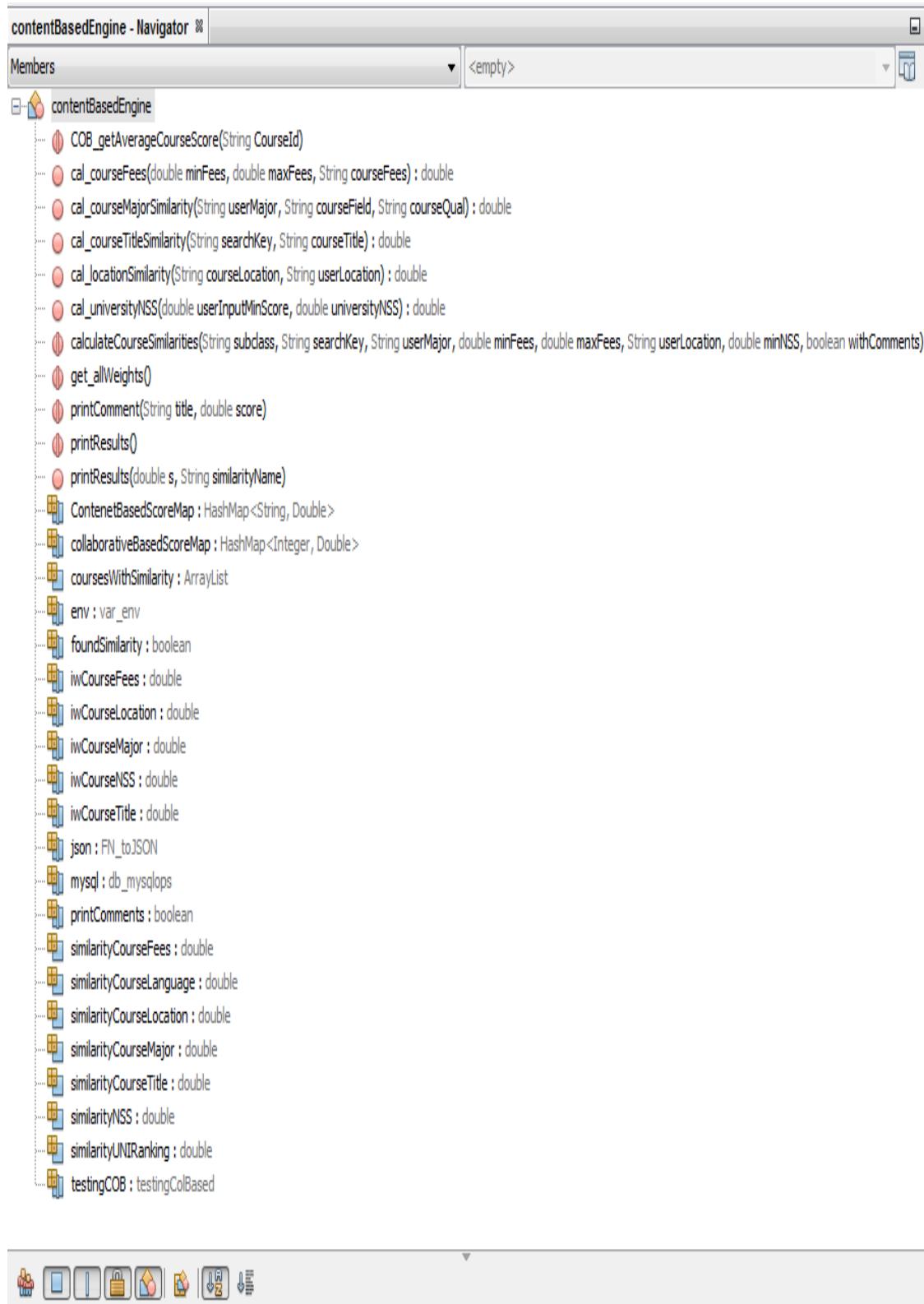


Figure B.7 Figure A.B. 8 CBF and CF functions and classes in the code

## Appendix C User and admin Interface pages

### 1. User interface main page



Figure C.1 Snapshot of the user interface main page

### 2. User registration page

The screenshot shows a web browser window with the URL [localhost:8080/Ontology/registration.html](http://localhost:8080/Ontology/registration.html). The page is titled "Registration" and has a sub-instruction "Build your profile with few steps!". It is divided into two main sections: "Personal Data" and "About your Education".  
**Personal Data:** Fields include First Name (Mohammed), Middle Name (Esmat), Last Name (Ibrahim), Gender (Male), Address (Kings road), City (Portsmouth), Email (mohammed3@gmail.com), Password (\*\*\*\*\*), and Mobile (07448038575). A validation message "mohammed3@gmail.com is a valid address" is displayed next to the email field.  
**About your Education:** Fields include Which Field? (Information technology), Field of study (Information technology), Main Subject Study (Computer Science), Major Subject (Artificial intelligent), Preferred Language (English), Current course level (BSc), User's Skills (Programming, machine learning), Interest Area (system experts), and a note about course enrollment. A "Submit" button is located at the bottom of the form.

Figure C.2 Snapshot of registration page in OPRCourse

### 3. Activities on the administrator page

localhost:8080/Ontology/administration.html#!/modifyClasses

Classes	Interest Areas	User Features Weights	Course features Weights
<p>View in which the Administrator will be able to Add new classes configuration or edit the current classes.</p> <p><a href="#">Show Classes</a></p> <p><a href="#">New Classes</a></p>	<p>View in which the Administrator can add Interest Areas to be available to the user while the registration process.</p> <p><a href="#">Show Interests</a></p> <p><a href="#">New Interests</a></p>	<p>This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.</p> <p><a href="#">Show Weights</a></p>	<p>This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.</p> <p><a href="#">Show Weights</a></p>
<b>Field of study</b> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text" value="Enter Field of Study's Name"/> </div> <p>Career name will be the Root element for classes and subclasses.</p> <p><b>Class Name</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text" value="Enter Class Name"/> </div> <p>Classes are the parent element for the subclasses, first child element for the career element.</p> <p><b>Subclass Name</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text" value="Enter Subclass Name"/> </div> <p>Subclasses are the last child in the configuration tree.</p> <p><a href="#">Submit</a> <a href="#">New</a></p>			

Figure C. 3 Dynamic configuration of course ontology

localhost:8080/Ontology/administration.html#!/viewConfig

Classes	Interest Areas	User Features Weights	Course features Weights	Languages	Course Levels	Manage Algorithms
<p>View in which the Administrator will be able to Add new classes configuration or edit the current classes.</p> <p><a href="#">Show Classes</a></p> <p><a href="#">New Classes</a></p>	<p>View in which the Administrator can add Interest Areas to be available to the user while the registration process.</p> <p><a href="#">Show Interests</a></p> <p><a href="#">New Interests</a></p>	<p>This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.</p> <p><a href="#">Show Weights</a></p>	<p>This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.</p> <p><a href="#">Show Weights</a></p>	<p>View/Edit languages reference.</p> <p><a href="#">Show Languages</a></p>	<p>Add/Remove Course Current level</p> <p><a href="#">Show Levels</a></p> <p><a href="#">Add Language</a></p>	<p>This is a very critical as from this view you can manage all the algorithm performance, content-based.collaborative-based weights</p> <p><a href="#">Order Levels</a></p> <p><a href="#">Show Levels</a></p>
<p>Level Name</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Content Based : Value =&gt; 50  Collaborative Based : Value =&gt; 30  University Rank : Value =&gt; 10  University NSS : Value =&gt; 10  KNN Scale Start : Value =&gt; 1  KNN Scale End : Value =&gt; 5  Course Fees Scale Start : Value =&gt; 8000  Course Fees Scale End : Value =&gt; 25000 </div>						

Figure C. 4 Dynamic configuration algorithms weights in OPCR

**Classes**

View in which the Administrator will be able to Add new classes configuration or edit the current classes.

[Show Classes](#)

[New Classes](#)

**Interest Areas**

View in which the Administrator can add Interest Areas to be available to the user while the registration process.

[Show Interests](#)

[New Interests](#)

**User Item Weights**

This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.

[Show Weights](#)

**Course Item Weights**

This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.

[Show Weights](#)

**Languages**

View/Edit languages reference.

[Show Languages](#)

[Add Language](#)

**Course Levels**

Add/Remove Course Current level

[Show Levels](#)

[Order Levels](#)

**Manage Algorithm**

This is a very critical as from this view you can manage all the algorithm performance, content-based,collaborative-based weights

[Show Levels](#)

Filter

Language ID	Language Name	
13	English	<a href="#">DELETE</a>
14	France	<a href="#">DELETE</a>
15	German	<a href="#">DELETE</a>
16	Arabic	<a href="#">DELETE</a>
17	Spanish	<a href="#">DELETE</a>

Figure D. 3 dynamic Configuration of Language feature

**Classes**

View in which the Administrator will be able to Add new classes configuration or edit the current classes.

[Show Classes](#)

[New Classes](#)

**Interest Areas**

View in which the Administrator can add Interest Areas to be available to the user while the registration process.

[Show Interests](#)

[New Interests](#)

**User Item Weights**

This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.

[Show Weights](#)

**Course Item Weights**

This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology.

[Show Weights](#)

**Languages**

View/Edit languages reference.

[Show Languages](#)

[Add Language](#)

**Course Levels**

Add/Remove Course Current level

[Show Levels](#)

[Order Levels](#)

**Manage Algorithm**

This is a very critical as from this view you can manage all the algorithm performance, content-based,collaborative-based weights

[Show Levels](#)

Filter

ID	Item Name	Item Weight		
1	mainClass	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
2	subClass	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
3	Class	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
4	region	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
5	Skills	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
6	interestArea	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>

**Add new item**

New item Name

New item weight

[Add](#)

Figure D.4 dynamic Configuration user feature weights

① localhost:8080/Ontology/administration.html#!/viewCourseWeights

The screenshot shows a web-based administrative interface for managing course features and weights. The top navigation bar includes links for 'Home', 'Ontology', 'Administration', 'Logout', and a search icon.

The main content area is divided into several sections:

- Classes**: View in which the Administrator will be able to Add new classes configuration or edit the current classes. Buttons: [Show Classes](#), [New Classes](#).
- Interest Areas**: View in which the Administrator can add Interest Areas to be available to the user while the registration process. Buttons: [Show Interests](#), [New Interests](#).
- User Features Weights**: This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology. Buttons: [Show Weights](#).
- Course features Weights**: This view for adding new item with a weight (percentage) which will be used to order the priority of data items for user's ontology. Buttons: [Show Weights](#).
- Languages**: View/Edit languages reference. Buttons: [Show Languages](#), [Add Language](#).
- Course Levels**: Add/Remove Course Current level. Buttons: [Show Levels](#), [Order Levels](#).
- Manage Algorithm**: This is a very critical as from this view you can manage all the algorithm performance, content-based, collaborative-based weights. Button: [Show Levels](#).

A central table displays course feature weights:

ID	Item Name	Item Weight	EDIT	DELETE
23	Course Title	15 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
24	Course Major	15 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
25	Course Location	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>
26	Course Fees	10 %	<a href="#">EDIT</a>	<a href="#">DELETE</a>

An 'Add new item' form is present at the bottom left:

New item Name:   
 New item weight:

[Add](#)

Figure D.5 dynamic Configuration course feature weights

## Appendix D OPCR experiment evaluation form

### OPCRA Framework evaluation Form

#### 7. Quality of Recommended Items

Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The items recommended to me matched my interests	<input type="radio"/>				
This recommender system gave me good suggestions.	<input type="radio"/>				

#### 8. Relative Accuracy

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The recommendations I received better fits my interests than what I may receive from a friend	<input type="radio"/>				

#### 9. Familiarity

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Some of the recommended items are familiar to me.	<input type="radio"/>				

#### 10. Attractiveness

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The items recommended to me are attractive	<input type="radio"/>				

#### 11. Novelty

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
This recommender system helped me discover a new course	<input type="radio"/>				
The items recommended to me are novel	<input type="radio"/>				

#### 12. Diversity

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The items recommended to me are diverse.	<input type="radio"/>				
The items recommended to me are similar to each other	<input type="radio"/>				

**13. Interaction Adequacy***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
This recommender system allows me to tell what I like/dislike.	<input type="radio"/>				
This recommender system allows me to modify my taste profile.	<input type="radio"/>				
This recommender system explains why the courses have recommended to me.	<input type="radio"/>				

**14. Interface Adequacy***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The information provided for the recommended items is sufficient for me to make a choose decision.	<input type="radio"/>				
The labels of this recommender system interface are clear.	<input type="radio"/>				
The labels of this recommender system interface are adequate.	<input type="radio"/>				
The layout of this recommender system interface is attractive.	<input type="radio"/>				
The layout of this recommender system interface is adequate	<input type="radio"/>				

**15. Perceived Ease of Use***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I found it easy to tell this recommender system what I like/dislike.	<input type="radio"/>				
I found it easy to modify my taste profile in this recommender system.	<input type="radio"/>				
I found it easy to make this recommender system recommend new items to me.	<input type="radio"/>				
I found it easy to inform the system if I dislike/like the recommended item.	<input type="radio"/>				

**16. Ease of Initial Learning***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I became familiar with this recommender system very quickly.	<input type="radio"/>				
I easily found the recommended items.	<input type="radio"/>				

**17. Perceived Usefulness***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
. This recommender system helped me find the ideal item.	<input type="radio"/>				
This recommender system influenced my selection of items.	<input type="radio"/>				
Using this recommender system to find what I like is easy	<input type="radio"/>				
Finding an item to course select with the help of this recommender system is easy.	<input type="radio"/>				

**18. control/Transparency***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I feel in control of telling this recommender system what I want.	<input type="radio"/>				
I feel in control of modifying my taste profile.	<input type="radio"/>				
Using this recommender system to find what I like is easy.	<input type="radio"/>				
I understood why the items were recommended to me.	<input type="radio"/>				
This system's recommender system seems to control my decision process rather than me.	<input type="radio"/>				

**19. Attitudes***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Overall, I am satisfied with this recommender system.	<input type="radio"/>				
. I am convinced of the items recommended to me.	<input type="radio"/>				
I am confident I will like the items recommended to me.	<input type="radio"/>				
This recommender system made me more confident about my selection/decision.	<input type="radio"/>				
The recommended items made me confused about my choice.	<input type="radio"/>				
This recommender system can be trusted.	<input type="radio"/>				

**20. Behavioral Intentions***Mark only one oval per row.*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I will use this recommender to find items to choose a university course.	<input type="radio"/>				
I will use this recommender again.	<input type="radio"/>				
I will use this recommender frequently.	<input type="radio"/>				
I will use this recommender rather than other recommenders.	<input type="radio"/>				
I will tell my friends about this recommender system	<input type="radio"/>				

**21. What aspects of this system were most useful or valuable?**


---



---



---



---



---

**22. How would you improve this system?**


---



---



---



---



---

**23. which features are important when you select university course***Tick all that apply.*

- Course title
- University Location
- Course Fees
- Course Mode
- University Rank
- University NSS score

# FORM UPR16

## Research Ethics Review Checklist

Please include this completed form as an appendix to your thesis (see the Research Degrees Operational Handbook for more information)



Postgraduate Research Student (PGRS) Information		Student ID:	445005
PGRS Name:	Mohammed Essmat Ibrahim		
Department:	School of Energy and Electronics Engineering	First Supervisor:	Dr Linda Yang
Start Date: (or progression date for Prof Doc students)	01/02/2015		
Study Mode and Route:	Part-time <input type="checkbox"/> Full-time <input checked="" type="checkbox"/>	MPhil <input type="checkbox"/> PhD <input checked="" type="checkbox"/>	MD <input type="checkbox"/> Professional Doctorate <input type="checkbox"/>
Title of Thesis:	An Ontology-based Hybrid Approach to Course Recommendation in Higher Education		
Thesis Word Count: (excluding ancillary data)	45286		

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

### UKRIO Finished Research Checklist:

(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: <http://www.ukrio.org/what-we-do/code-of-practice-for-research/>)

a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>
b) Have all contributions to knowledge been acknowledged?	YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>
e) Does your research comply with all legal, ethical, and contractual requirements?	YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>

### Candidate Statement:

I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)

**Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):** 20EC-62C5-E27D-2F06-3393-6F12-C17F-B1BA

If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:

Signed (PGRS):

Date: 30-01-2019

