

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Ontology-based Personalised Course Recommendation Framework

MOHAMMED E. IBRAHIM^{1,2}(Member, IEEE), YANYAN YANG^{1,2}, DAVID NDZI³,
GUANGGUANG YANG¹, MURTADHA ALMALIKI¹

¹School of Engineering, University of Portsmouth, Portsmouth PO1 3DJ, U.K

²School of Computing, University of Portsmouth, Portsmouth PO1 3HE, U.K

³School of Engineering and Computing, University of the West of Scotland,
Paisley, Scotland, U.K

Corresponding author: Mohammed E. Ibrahim (e-mail: mohammed.ibrahim@port.ac.uk).

This work was supported in part by the Higher Committee Education Development in Iraq through the Scholarship Program of the Iraqi Government, and in part by the University of Portsmouth.

ABSTRACT 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. A personalised recommendation system can be an effective way of suggesting relevant courses to prospective students. This paper introduces a novel approach that personalises course recommendations that will match the individual needs of users. The proposed approach developed a framework of an ontology-based hybrid-filtering system called OPCR. This approach aims to integrate information from multiple sources based on hierarchical ontology similarity with a view to enhancing efficiency and user satisfaction and to provide students with appropriate recommendations. OPCR combines collaborative based filtering with content-based filtering. It also considers familiar related concepts that are evident in the profiles of both the student and the course, determining the similarity between them. Furthermore, OPCR uses an ontology mapping technique, recommending jobs that will be available following completion of each course. This method can enable students to gain a comprehensive knowledge of courses based on their relevance, using dynamic ontology mapping to link course profiles and student profiles with job profiles. Results show that a filtering algorithm that uses hierarchically related concepts produces better outcomes compared to a filtering method that considers only keyword similarity. In addition, the quality of the recommendations improved when the ontology similarity between the items' profiles and the users' profiles were utilised. This approach, using a dynamic ontology mapping, is flexible and can be adapted to different domains. The proposed framework can be used to filter items for both postgraduate courses and items from other domains.

INDEX TERMS Information Overload, Recommendation Systems, Course Recommender system, Ontology, Education Domain

I. INTRODUCTION

Finding 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 real challenge [1]. We have used the term "course" in this paper to refer to any program of study such as undergraduate, postgraduate and so forth. 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 the one that is most appropriate for the student [2]. Furthermore, even though some course titles are similar, they can lead to a different career path [3]. Studies have shown that, naturally, the students' choices are influenced by their background, personal interests and career interests [4]. Researchers Gordon and Cuseo found that three out of every four students were uncertain or tentative about their career choice at the time of college entry [5].

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 [6]. However, simply because more course information is now provided on university websites, this does not automatically

mean that students possess the cognitive ability to evaluate them all [1]. Instead, they are confronted with a problem that is termed “information overloading” [7].

Artificial intelligence methods developed at the beginning of research are now being applied to information retrieval systems. Recommended systems provide a promising approach to information filtering [8] as they help users to find the most appropriate items [9]. Based on the needs of each user recommendation system, a series of specific suggestions will be generated [10]. Recommendation systems are widely classified into three main techniques in the literature: collaborative-based filtering (CF) [11], [12]–[15], content-based filtering (CBF) [16] and hybrid filtering [17]–[21].

There are many online systems currently available that can be used to find and search for courses [22], which use tools based on the users’ prior knowledge of the courses [19], keyword-based queries [23], [24] collaborative filtering based [25] [26], data mining and association rules based [19], [27] and content-based filtering models [28]. Despite the high impact of the course recommendation system and how useful it is, there are certain significant limitations, such as:

- Models 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 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 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.

Through 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 [6]. By using this

ontology, the user will be able to gain precise knowledge about the course [22]. We have been able to build a relationship between the relevant information available through 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 [29]. The use of semantic descriptions of the courses and the students’ profiles allows there to be both qualitative and quantitative reasoning regarding the matching, as well as the required information about the courses and the student’s interests which is necessary in order to refine the process of deciding which course to select.

A novel hybrid filtering is proposed in this study, 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. This problem is related to the sparsity of information that is available (i.e. for users and items) in the recommendation filtering algorithms [30]. The proposed approach uses 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 that is based on ontology, item ontology, the semantic similarity between two ontologies and the proposed OKNN algorithm is 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. In order to ensure the measurement of semantic similarity is more accurate, a heuristic method is used in the CBF. 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.

This paper is structured as follows. In Section 2, we discuss related work that is relevant to this study. Section 3 presents the proposed methodology with all of the process functions. Section 4 presents the implementation and evolution of the methodology, Section 5 describes the discussion results and finally, Section 6 includes the conclusion and recommendations for future work.

II. RELATED WORK

A recommender system is a tool that provides personalised recommendations for those items that are most likely to be relevant and interesting to a user in order to help him/her to find the most useful items [13], [31]. Recommended items can be any products, services, books, news or information in a given application domain. Recommender systems have been applied in different domains, including the traditional e-commerce domain and, remarkably, in emerging domains such as education and engineering [13], [32], [33].

Recommendation systems have more recently offered personalised and more relevant recommendations. The personalised approach is achieved by using information that is found in certain situations, such as studying various objects, the context and areas of interest, location and careers [34]–[36]. For instance, courses that are recommended to a student who wishes to work in IT and who searches for “business information” will differ from those that are recommended to a student who aims to become an academic member of staff in the same area, since their requirements and level of education will be different. It is treated as contextual data that is a significant source of the accuracy of the recommendations [37], [38].

Various approaches are contained within the recommendation system. The main approaches are content-based filtering (CBF), collaborative-based filtering (CF) and hybrid-based filtering [16], [19], [39]. CBF attempts to recommend items that are related to those which a given user has preferred in the past. In CF, however, the system identifies users whose preferences are similar to those of the given user and suggests items they have favoured. A hybrid recommender system is one that combines two or more recommendation approaches to achieve a better performance with fewer disadvantages than an individual approach. CF is combined, most frequently, with certain other techniques in an attempt to avoid the cold start problem. In the following sub sections, we focus on literature using recommendation systems in the education domain and how the use of ontology will improve the quality of the recommendations.

A. RECOMMENDATION SYSTEM IN EDUCATION DOMAIN

A significant number of recommender systems have been proposed in the education domain, as well as in teaching and academic advising. 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. Sandvug and Burke presented Academic Advisor Course Recommendation Engine (AACORN) that used a case-based reasoning approach, which utilised knowledge that had been acquired from previous cases in order to solve new problems [40]. Their system used both the course histories and experience of past students as the basis of assisting students in course decision making.

At the same time, it was noticed that the future career of students is an essential factor which can influence their decision to choose a particular course [41]. Farzan and Brusilovsky proved this by using a reported course recommendation system that was based on an adaptive community [3]. They employed a social navigation approach to analyse the students’ assessment of their career goal in order to provide recommendations for courses. The primary idea 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 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 [21]. The challenge for recommender systems, therefore, is to better understand the student’s interest and the purpose of the domain [1]. An association mining based recommender has been developed for recommending tasks that are related to learning, and are most suitable for learners based on the performance of the targeted student and other students who are similar to them [27]. A course recommendation system has been proposed that would check how similar university course programmes are to the students’ profiles.

The proposed framework is a comprehensive one that combines CBF and CF with an ontology technique in order to overcome the overloading information problem. It does this by using a similar hierarchical ontology to map the courses profiles with the user (student) profile. The new approach develops two new methods to extract and integrate data from multiple sources and then link them. 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 there is between the users’ profiles by measuring the user rates for each item. The proposed recommender system is used to work out the hierarchy ontology similarity there is between the item profiles and users’ profiles before the student enrolling in the research program chooses courses to match his/her requirements.

B. ONTOLOGY BASED RECOMMENDER SYSTEM

The original definition of “ontology” in computer science was provided by Gruber [42] as being “explicit specification of a conceptualisation”. An ontology is used to represent an area of knowledge that formally describes a list of terms. Each of these items represents an important concept, such as the classes of objects and the relationships that exist between them [29]. Ontologies provide formal semantics that can be used to both process and integrate a range of information on the internet. Modelling information is one of the main goals of using ontologies [43]. The authors in [44] reported that ontologies are concept properties, disjointness statements, value restrictions and specifications of logical relationships between objects. Ontologies provide a tool for the formal modelling of the structure of a system, which is based on the relationships that emerge from its observation.

The term taxonomy (topic hierarchy) has been used when the ontology contains only “IS-A” relationships. The use of the word ‘ontology’ is usually restricted to systems that support a rich variety of relationships between concepts, including logical propositions that formally describe the relationship. Many ontology classifications have been established [45]. For instance, ontology can refer to specific domains that may provide conceptual modelling of a particular domain.

Various ontology-based recommendation approaches have been developed by using a variety of different methods [46]. Furthermore, the concept of the semantic web is used to improve e-learning. In [47], Yang et al proposed a semantic recommender system approach for e-learning use 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 [10] 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 user information from the past is insufficient [30]. 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 [48]. Ontology structure significantly improves the ontology structure, which can lead to increased accuracy [49]. For instance, all of the “IS-A” 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. Consequently, this affects how accurate the recommender system is in finding similar items or users. To avoid this problem in our system, we will invite new users to complete their profile by providing their personal information and preferences and by responding to certain questions. We will then create the user profile model based on the user ontology model.

The main contributions of this work are the following:

- Develop a comprehensive framework that combines CBF and CF with an ontology technique in order to overcome the overloading information problem. This is achieved by using a similar hierarchical ontology to map the profiles of the courses with the user (student) profile.
- Develop a new approach to extract and integrate data from multiple sources and then map them. 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 there is between the users’ profiles by measuring the user

rates for each item. The proposed recommender system is used to work out the hierarchy ontology similarity there is between the item profiles and the users’ profiles before the student enrolls on the research program and chooses courses to match his/her requirements.

III. PROPOSED SOLUTION

A hybrid recommender method based on ontology has been proposed in this work. The method firstly aims to extract and integrate information from multiple sources based on ontology. The information sources are classified into three primary sources; course information sources, student information sources and career information sources. Integrating information using ontology will obtain an optimal result. 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 recommend an appropriate recommendation to the users, we have combined two main filtering approaches, CBF and CF, and thus the result is a combination of memory-based and model-based methods. In the CF, several techniques, such as user profiling that is based on ontology, item ontology and k-NN, are used to overcome the information overload problem and improve scalability and accuracy.

On the other hand, item-based ontology and semantic similarity are applied in the content-based filtering to solve the new user issue and to also 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 gives their feedback on the recommendation list and carries out a re-ranking. User feedback has been used to evaluate the system and improve its accuracy, as is shown in greater detail in the evaluation section. This work aims to increase the accuracy and performance of the recommender system by combining the hybrid method (CBF and CF) with enhanced ontology.

A. FRAMEWORK OVERVIEW

The proposed ontology-based personalised course recommendation framework (OPCR) is focused 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.1, OPCR consists of four main layers. The first layer is data gathering, which consists of all the information resources and the data collection model. This is used to extract useful information from multiple sources. The second layer is the database, which is used to store all of the items and user information. The middle layer is the core functional part, which includes the ontological data model and the recommender engine model. We will explain each model in detail in the following sections. The final layer is a user application layer that consists of the user interface model, which is responsible for user interaction with the framework,

for searching items and for giving feedback on the recommendation list. Every layer and model in the framework both links and interacts with the others, based on the input and output of each one. Our framework comprises the following steps:

- (1) Extract all the useful information for the system from multiple sources.
- (2) Build the courses' profiles by extracting all the useful information regarding course features and sorting 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. We have identified different user attributes which can be used to profile the student into our system as well as the user ratings of the recommended courses.
- (4) Build dynamic ontology mapping in order to link the user profile and 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 user feedback. The purpose of each of these components is explained in the following sections.

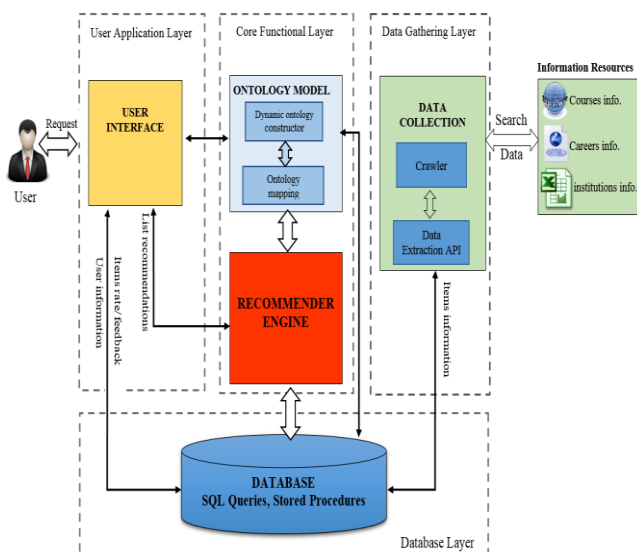


FIGURE 1. OPRC main architecture

B. DATA GATHERING

As it was decided that a content-based recommender system technique should be the primary approach for the provision of recommendations, there are different formats of information that need to be gathered to support this system. Fortunately, all of these are available through information sources which are publicly available, either through websites in HTML format, such as the universities' websites for course information and recruitment websites for career

information, or Microsoft Excel documents that have been uploaded to the internet, such as statistical information regarding the reputation of educational institutions, for example the NSS score 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. It was a time-consuming task to obtain information about each course from all the universities' websites as each university publishes its course information in different formats. Extracting precise information from various websites is always a challenging task in the domain of information engineering so we customised a web crawler that browses the web page automatically. It 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 feature values. Each extracted feature value belongs to one of the features that we have used in this paper. Five features of the courses are marked in this study: course title, course major subject, course fee, university location and the language of the course. On the other hand, the feature that has been constructed in the user ontology is based on the feature in item ontology. The implicit information, such as the 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.

C. CORE FUNCTIONAL

This section is the most important part of the framework and it consists of two models. Firstly, the ontology model, which includes construction dynamic ontologies for the user and the items that map these ontologies in order to gain a comprehensive knowledge of the recommendations. After building the ontologies and mapping them, this will be used as an input in the recommender engine. The recommender engine model is the second model in the layer. We have combined both CBF and CF filters to recommend items to users and utilised ontology in order to enhance the performance of the recommender engine (see section D and E for more details).

Ontologies are used in the proposed approach 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 representation, the term ontology refers to both the formal and explicit descriptions of the domain concepts [1]. These are frequently conceived as a set of entities, relations, functions, instances and axioms [7]. 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 the course content features based on their semantics. In addition, the hierarchical structure of the

ontologies allows the developers to reuse the domain ontologies (for example, in computer science and programming language)[50] in order to describe the learning fields and to build a practical model without the need to start 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 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 created significantly helps to reduce information overloading.

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 the dynamic from classical physics [43]. There are generally several ways to make a given static ontology become a dynamic one; it simply depends on what we want to define 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 improperly due to fewer terms being classified because of the limited and static nature of the classifiers. To overcome this, we propose using an ontology-based model 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 [44]. 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. Fig.2.

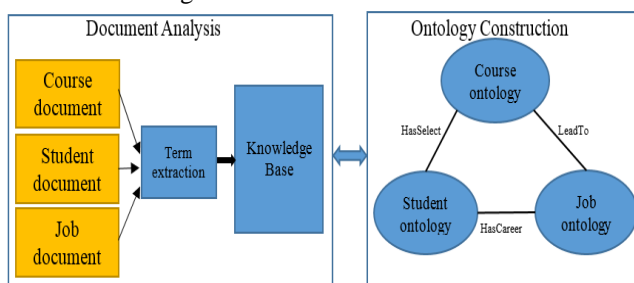


FIGURE 2. Dynamic Ontology Construction

There are many existing methods of constructing ontologies available. In the present work, we follow the ‘‘Ontology Development 101’’ approach developed by Natalya Noy and Deborah McGuinness [51]. The language used to write the ontology is the OWL 2 Web Ontology Language [36] and the protégé tool (Version 5.2) [52] 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 proposed work, 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, we define a property called *HasSelected*, which is used to represent this relationship. Data properties link an individual to a data literal, such as a student’s ID.

We 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 to a user according to his/her profile. In addition, in this work we have focused mainly on the recommendation of courses 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.

2) Course Ontology

Identifying different attributes is necessary for course profiling[31]. In order to construct a course ontology, we need to identify factors that most influence a student when they make a decision in choosing a university course. These

factors then become the main classes of the ontology. We carried out a survey of students at the University of Portsmouth to discover 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 to choose a university course and were then asked to rank these 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 have been summarised in Table 1.

TABLE 1. Factors and keys constituent elements for selecting university courses

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

A course programme’s title, fees, location and prominence were all factors that appeared to be the most important when the students determined their choice of university for higher education (HE) study. The students chose computer sciences and business management programmes, although some differences in the prioritisation of elements within the broad factors were observed. 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 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 the details regarding course information appear to exert the most considerable influence on the students’ choice of university course programme.

The factor that was uppermost in the students’ decision-making frameworks 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 university’s NSS score. This information is used for knowledge discovery at a later stage of the user profiling process. In Fig.3 the main classes and subclasses of course ontology are shown with instances.

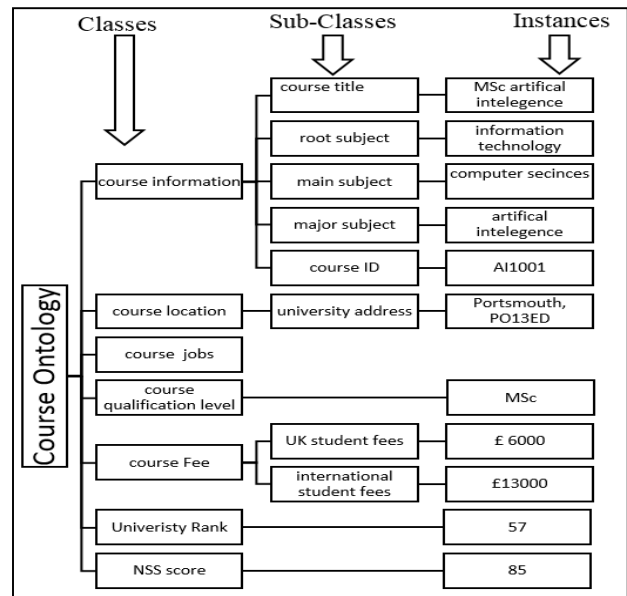


FIGURE .3 Course ontology Structure

The course profile attributes will match the user profile feature through the 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 shows the graphical representation of the course ontology in the protégé environment.

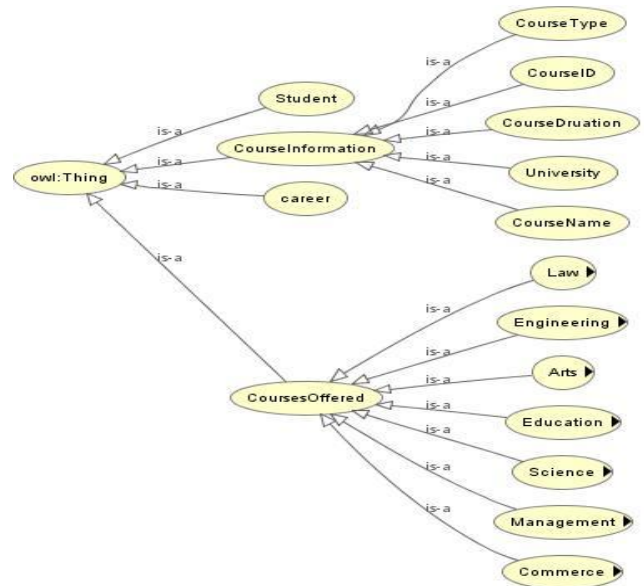


FIGURE 4. Graphical Representation of the Course Ontology

2) Student Ontology

Firstly, we need to model the student profile before recommending the appropriate course. The user profile consists of two main parts. The first part is the personal attributes and education attributes of the user and the second is the user's rating of the previously recommended course. The personal 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, as shown in Fig. 5 and in Fig.6, the graphical representation of student profile ontology in protégé environment is shown.

Therefore, in this paper, a student profile can be formally defined as Formula (1) and Formula (2):

$$U = \{ a_1, a_2, \dots, a_n \} \quad (1)$$

Where U is the user/student, a_i represents the users and i th attributes.

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

$$U_r = \{ u, r \} = \{ a_1, a_2, \dots, a_n, r \} \quad (2)$$

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 user's interests. 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 faculties 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 graduation degree, the system can recommend any postgraduate course and if the student has a postgraduate degree, either a research course or a PhD can be selected, depending on the faculty. The proposed approach conducts an entrance test as an eligibility criterion for admission into the undergraduate and postgraduate engineering courses.

In the proposed system, there are three ontologies: course ontology, student profile ontology and job 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 files. Unstructured text documents include four processes: data pre-processing, concept clustering, context extraction and local ontology construction. For more information about local ontology construction from the unstructured text, see [1].

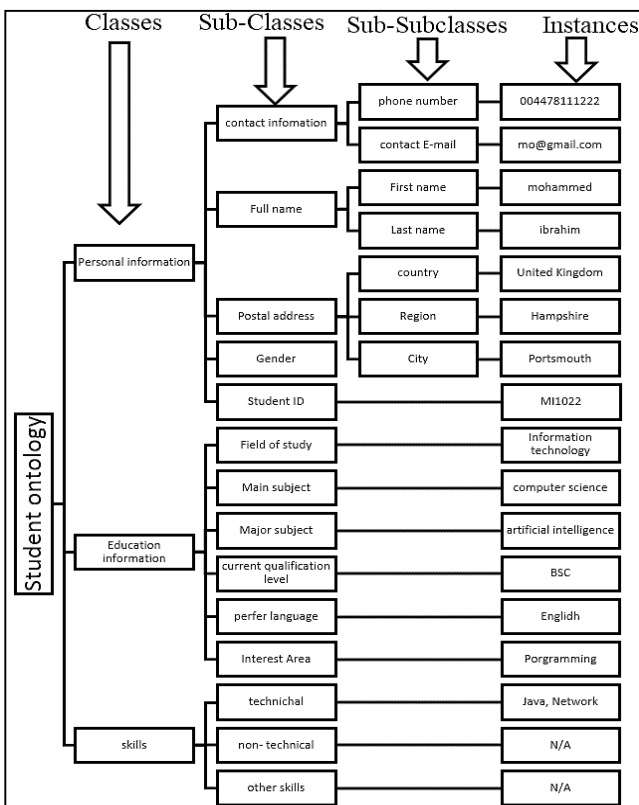


FIGURE 5. Student Ontology Structure

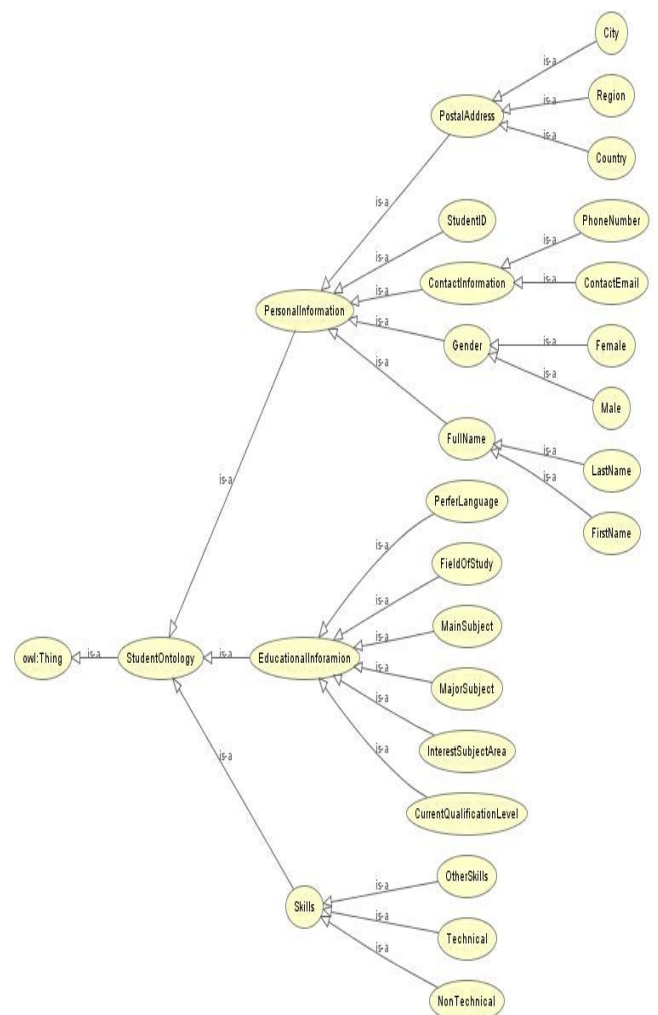


FIGURE 6. Graphical Representation of the Student Ontology

3) Job Ontology

A student's future career is an essential factor that can influence their decision making when they are selecting a university course [3]. Constructing a job ontology is vital if a student is to understand the attributes of the job. This is extracted from a recruitment 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.7.

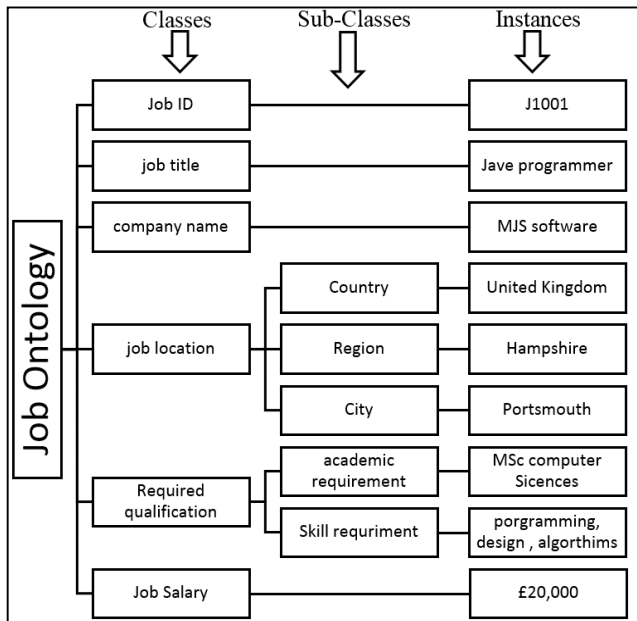


FIGURE 7. Job Ontology Structure

There is also a graphical representation of the job ontology in protégé environment in Fig.8.

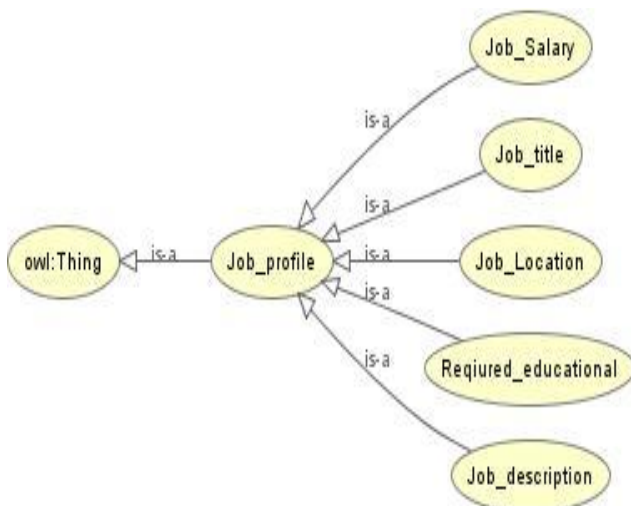


FIGURE 8. Graphical Representation of the Job Ontology

E. RECOMMENDER ENGINE

After constructing the ontology models, in this section we now discuss the recommender engine. We used a hybrid method which combined the CBF and CF filtering approaches with supporting ontology model mapping, and this is the core component of the framework. In the following sections, we explain in detail how each element of the hybrid approach works

1) CBF METHOD

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, we need to generate a vector for the features of both the item and the user. According to the course ontology model, the main classes are used as the feature of the item vector. The features include the course title, the major subject of the course, the course fee and the institution's location. A constant weight has been adjusted for each of these features. These are 15%, 15%, 10%, 10%, respectively. An additional feature that was used in the CBF filtering to recommend the more relevant course was the university's reputation and its NSS score. The weight assigned to each additional feature was 10% and 10%, respectively in the final scoring function.

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.

Moreover, the matching similarity has been used to compute the similarity between the user location and the location of the university that provides the courses. Additionally, we have matched the user's city with the regions of the universities in order to obtain more results. The cities are as classified by the United Kingdom. These are based on 12 regions and each region is formed of many cities. For example, the South East includes Portsmouth, Southampton and Kent amongst others. We have used a different type of similarity in the CBF approach, such as cosine similarity, matching similarity and normalisation similarity, depending on the nature of the feature. This is as follows:

- Used cosine similarity to calculate the course title and major course subject, according to the formula (3)

$$Sim(I_{f_a}, I_{f_b}) = \frac{I_{f_a} \cdot I_{f_b}}{\|I_{f_a}\| \times \|I_{f_b}\|} \quad (3)$$

Where I_{f_a} , I_{f_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):

$$FS(U, C) = \frac{F_{umax} - F_c}{F_{max} - (F_{min} - 1)} \quad (4)$$

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, we also matched the user's city 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:** we calculated the ranking attribute in the user query and course profile, according to the formula (5)

$$RS(U, C) = \frac{R_{max} - R_c}{R_{max} - R_{min}} \quad (5)$$

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 which provides 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 (6) has been used:

$$NS(U, C) = \frac{U_N - (N_{min} - 1)}{N_{max} - (N_{min} - 1)} \quad (6)$$

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

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, we explain how the CF works within the framework and how using the ontology-enhanced CF performs to find the most similar users to the active user. The most important aspect of the CF is how to measure the similarity between the active user and the 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.

• 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 the item in the CBF case. In the proposed work, a new method has been used 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 system has ontology support from the user history similarity that enables it to calculate the similarity between the target user and the other users in the system, according to the formula (7). The user similarity value range will be between (0, 1) and the weight for each part 50%.

$$US(U_a, U_n) = \text{ontology similarity} + \text{recommendation history similarity} \quad (7)$$

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:
Level1 (major subject, main subject, the field of study)
Level2 (interest area)
Level3 (user location)
Level4 (user skills), as shown in Fig.9.

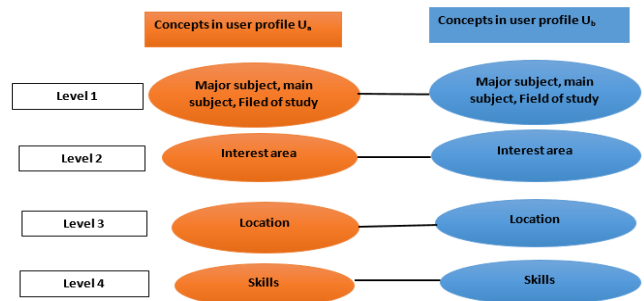


FIGURE 9. Hierarchical matching and matching parameters

To compute the similarity between each level of the ontology, we need to adjust the weight of each level, 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 School of Computing and School of Business at the University of Portsmouth. The results of the survey 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%)
- Level4 (5%)

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.(8):

$$OS(U_a, U_b) = \sum_{l=1}^n L_m \quad (8)$$

Where:

OS= Ontology similarity

N = number of levels in the ontology

L_m = 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 is most widely used in this area [18]. The similarity between the users' recommendation histories has been computed according to Eq. (9) and Eq. (10), as follows:

$$Sim(U_a, U_b) = \frac{\vec{U}_a \cdot \vec{U}_b}{\|\vec{U}_a\| \times \|\vec{U}_b\|} \quad (9)$$

$$Sim(U_a, U_b) = \frac{\sum_{p \in P} U_a \cdot U_b}{\sqrt{\sum_{p \in P} (U_a)^2} \sqrt{\sum_{p \in P} (U_b)^2}} \quad (10)$$

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. Following this, we need to normalise the value 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.

• 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, we proposed a new algorithm, 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. (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, we needed to rank the users' similarity score. A common rate problem we 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:

$$\text{Average weight score} = \left(\frac{\text{ARW} \cdot C \cdot (\text{KNNW} - \text{Omax} \cdot K)}{\text{KNNW}} \right) + \text{Oc} \cdot K / 100 \quad (11)$$

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

The proposed method improves the scalability and accuracy, leading to an improvement in the performance of the algorithm. We present the steps of this algorithm as follows:

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

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.

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, the other factor will be added to the final score as well, such as the university ranking and the NSS score as shown in the Eq. (12). The value of the final score function similarity should be between the range (0-1). The weight percentage for each part in FCF (CBF, CF,

university rank, NSS score) is 50%, 30%, 10%, 10%, respectively.

$$\text{Final Scoring Function (FSF)} = (\text{CBF} * (50\%)) + (\text{CF} * (30\%)) + (\text{university rank} * (10\%)) + (\text{NSS score} * (10\%)) \quad (12)$$

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:	If the user chooses any of the recommended courses then
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:	end if
11:	Return the refined recommendations to the user

IV. EXPERIMENTAL STUDY

An experimental prototype system has been designed based on the OPCR 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.

A. DESCRIPTION OF THE EXPERIMENT

We 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 2 shows the number of students from each level.

TABLE 2. Number of participants and level of study

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

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 [53] and the commercial Amazon system [11]. 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.

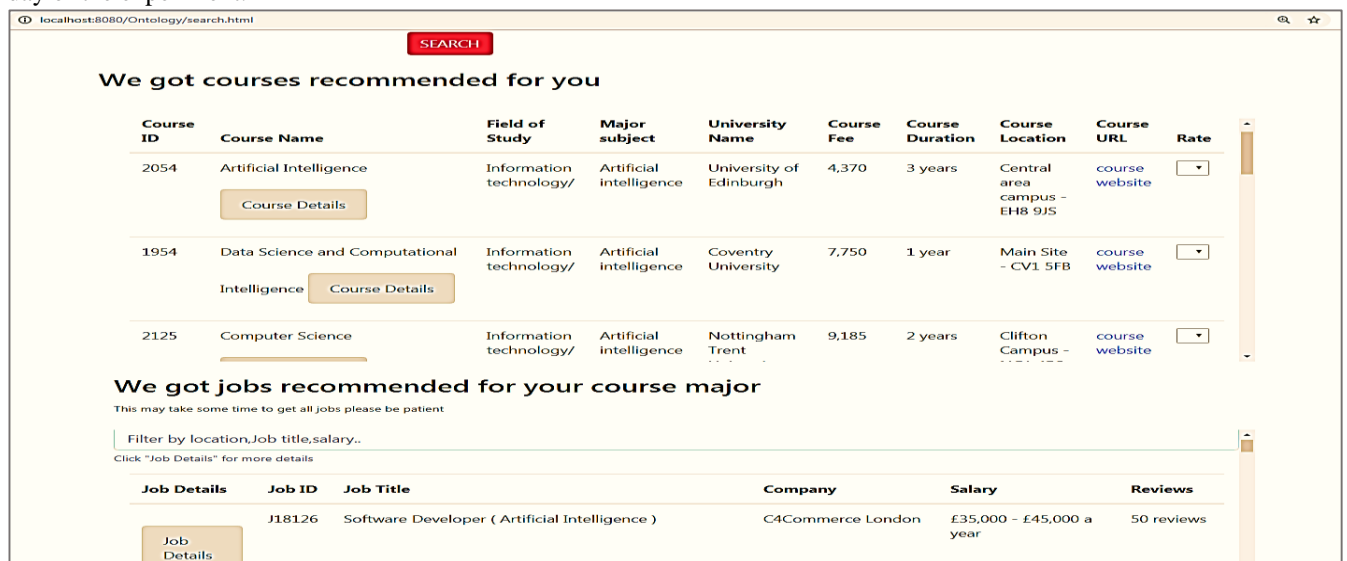


FIGURE 10. Course and job recommendations

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.

B. DATA SOURCE AND CONFIGURATION

In this study, 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. 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 that related to CS and BAM courses were extracted.

C. 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 [54]. The approaches will be discussed in detail in the following subsections.

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 [55]. 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.

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 [56]. 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.

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 [57]. 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. (13):

$$RC = \frac{\sum_{u \in u_{TestSet}} \frac{1}{Ku} \sum_{i=1}^{Ku} \frac{pi}{Cu}}{|u_{TestSet}|} \quad (13)$$

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 $|u_{TestSet}|$ 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 3, 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:

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

TABLE 3. Example of Recovery Metric

Ranking list	User 1		User 2		User 3		User 4		User 5	
	Course	R	Course	R	Course	R	Course	R	Course	R
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

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) [57]. 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. (14) and Eq. (15).

$$\text{Average Percision (AP)} = \frac{1}{M} \sum_{i=1}^n \text{rel}(k) \times P_{\text{rec}} @k \quad (14)$$

$$\text{mean Avarage Precision (MAP)} = \frac{1}{M} \sum_m AP_m \quad (15)$$

Where

M: the total number of relevant documents

n: The list length

rel (k): 1 if relevant, otherwise 0

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

m: number of queries

According to the example in Table 4, we have five users who received a list of recommended courses and they rated (R) this based on their interest. We used Eq. (14) to obtain the average precision for each user as the following:

TABLE 4. Example of Accuracy of list relevance Metric

User 1		User 2		User 3		User4		User 5	
Course Rec. List	R	Course Rec. List	R	Course Rec. List	R	Course Rec. List	R	Course Rec. List	R
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

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

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

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

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

$$\text{average precision for user 5} = \frac{\frac{1}{1}}{1} = 1$$

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. (16) as:

$$P(N) = \frac{\sum_{u \in u_{\text{TestSet}}} P_u(N)}{|u_{\text{TestSet}}|} \quad (16)$$

According to the example in Table 5, the precision will be as the following:

$$P(N) = \frac{\frac{3}{5} + \frac{2}{5} + \frac{3}{5} + \frac{2}{5} + \frac{1}{5}}{5} = 0.44$$

TABLE 5. Example of the percentage of relevant items to user u

Ranking list	User 1		User 2		User 3		User4		User 5	
	Course	R	Course	R	Course	R	Course	R	Course	R
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

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 [55]. 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. (17)

$$r = 1 - \frac{6}{n(n^2 - 1)} \sum_{i=1}^n (x_i - y_i)^2 \quad (17)$$

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:

TABLE 6. Example of Represent System Ranking and User Ranking case1

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

$$r = 1 - \frac{6}{5((5)^2 - 1)} ((1 - 2)^2 + (2 - 1)^2 + (3 - 5)^2 + (4 - 3)^2 + (5 - 4)^2) r = 0.6$$

The second case is for user U2 as shown in Table 7. 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:

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

$$r = 0.9$$

TABLE 7. Example of Represent System Ranking and User Ranking case2

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

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.

D. 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, as shown in Fig.11, 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 was 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 result, shown in Fig.12, was that the proposed approach algorithms worked far more precisely than the traditional one. Moreover, when we compared the proposed approach with a current course finder system, 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.13.

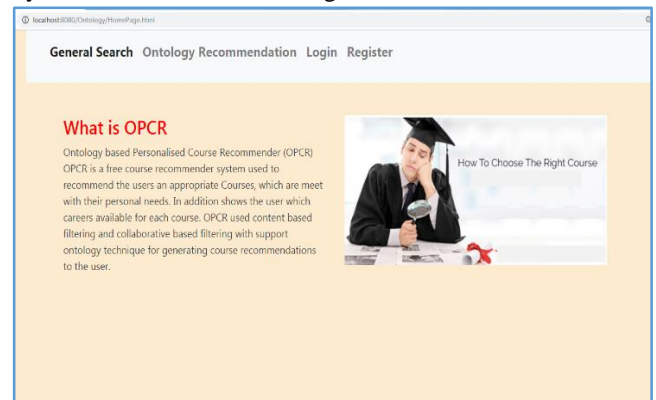


FIGURE 11. OPCR User Interface

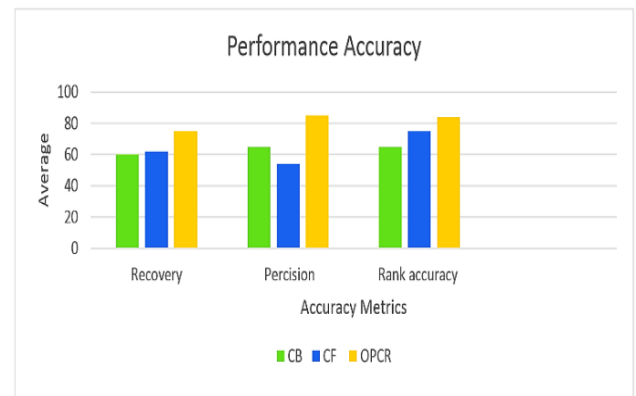


FIGURE 12. Comparison between POOCR and (CBF, CF) performance metrics

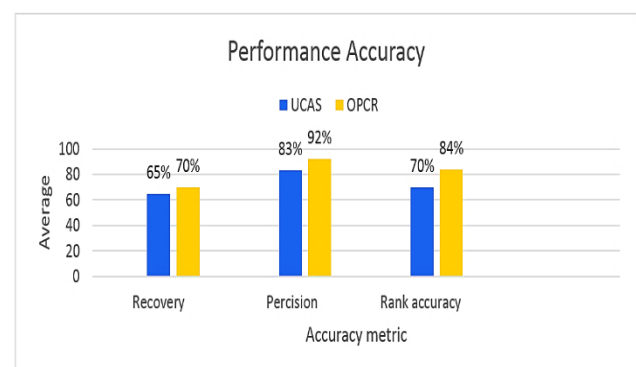


FIGURE 13. Comparison between POOCR and UCAS performance metrics

In contrast, we used a questionnaire 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 [58]. 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. 14. 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.

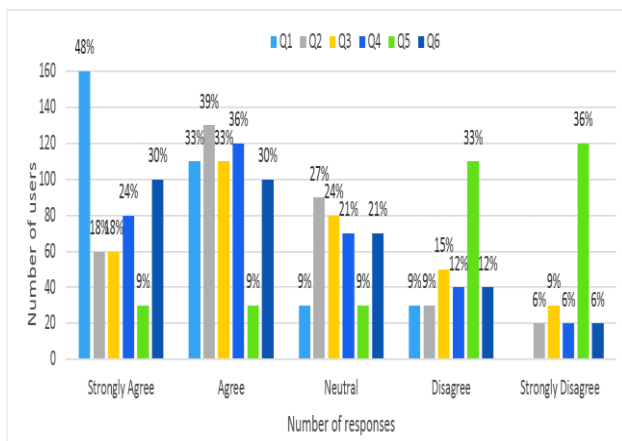


FIGURE 14. Attitudes Questions

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. 15 and more than 60% of the users were satisfied with the recommended courses regarding how the recommended course matched with users' interests.

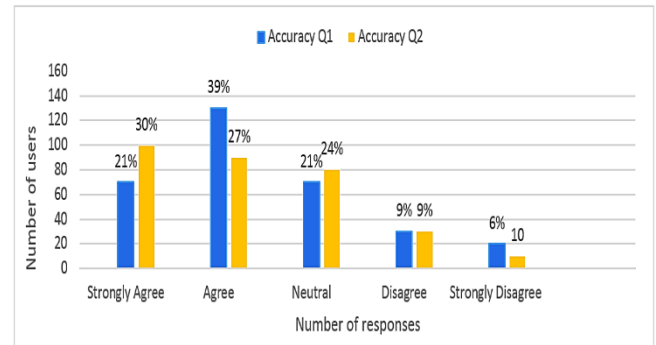


FIGURE 15. Accuracy Questions

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 65% had obtained recommendations which included some familiar items, and 35% of the users said the results included new items, as shown in Fig. 16.

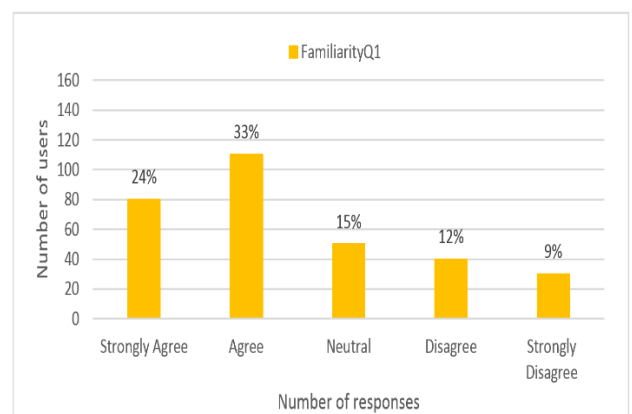


FIGURE 16. Familiarity Question

1.3. Novelty

Novelty is one of the important indicators of user satisfaction as it helps users in the decision-making process[59]. 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 as shown in Fig. 17,

Q1 The items recommended to me are novel
Q2 This recommender system helped me discover a new course

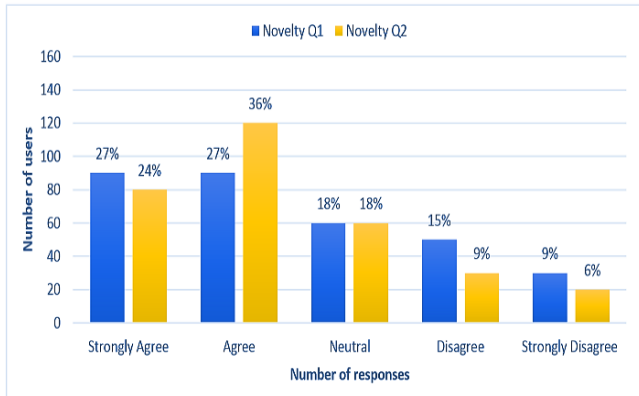


FIGURE 17. Novelty Questions

1.4. Diversity

The course domain in the recommendation system is different from that of other domains, such as news and movies [60]. 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.18 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

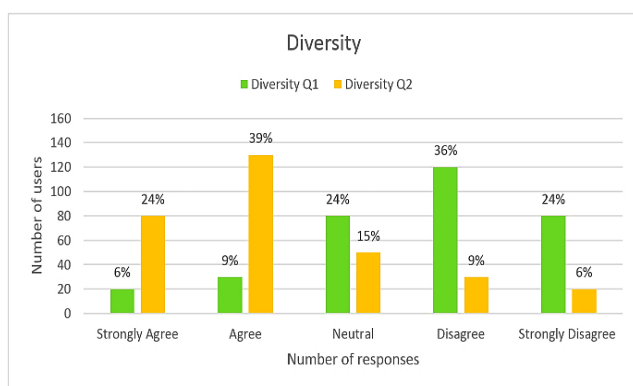


FIGURE 18. Diversity Questions

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.19.

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.

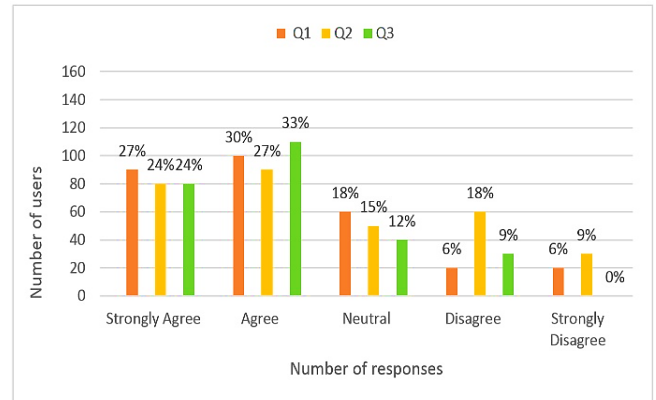


FIGURE 19. Interaction Adequacy Questions

V. CONCLUSION AND FUTURE WORK

Searching and finding an item that is relevant to the user is a huge challenge. Choosing a higher education course at university is a massive decision for students. The recommendation system in education plays a vital role in overcoming the problem of information overloading, and helps the students to find relevant and useful courses from a large number of online course resources that are available on the internet.

The current approaches to filtering have many limitations. To generate a comprehensive knowledge of the recommended items, information from multiple heterogenic sources needs to be mapped and linked. This paper proposes a novel approach to the recommendation system, which combines CBF and CF, supported by ontology similarity. OPCR algorithms are used to recommend university courses to a target student based on the user's interest and the choices made by similar students. The experiments showed that ontology matching is a desirable tool for making a recommendation to a target student, and it can be seen that the proposed approach can obtain better results, including greater user satisfaction and accuracy, than other approaches. Furthermore, the proposed approach can help to combat information overloading and the problems faced by new users by using ontology similarity between the users' profiles. Furthermore, using the ontology-based integration approach to integrate data from multiple heterogeneous sources will help the system to provide a comprehensive recommendation to users.

Throughout the experiments, we noticed that building a new hybrid recommendation system which combines CF and CBF utilising ontology improves the information

overloading problem. Moreover, the ontology mapping and recommendation filter algorithms were incorporated to improve the accuracy of the recommendation and increase the user stratification of the recommendation. In addition, this approach improves the new user problem in the CF by incorporating ontology similarity into the proposed method. It was found that using dynamic ontology mapping to link the course profiles and student profile with the job profile helped to provide comprehensive knowledge about the course that was not only more relevant to the student oncology-based course recommender system, but also successfully brought a new dimension of ontology domain knowledge about the student and the course resources into the recommendation process.

In future, we will enrich our repository by absorbing more course and user information and heterogeneous data sources. In addition, we plan to incorporate additional user contexts, e.g., available student behaviour, learning style and learning interests into the recommendation process in order to make the system more comprehensive and intelligent. We may employ more feedback information from students for effective courses and improve the student model based on students' feedback and consider more aspects and techniques related to recommender systems. We plan 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 our proposal.

REFERENCES

- [1] M. E. Ibrahim, Y. Yang, and D. Ndzi, "Using ontology for personalised course recommendation applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10404, pp. 426–438, 2017.
- [2] R. G. Apaza, E. V. Cervantes, L. C. Quispe, and J. O. Luna, "Online courses recommendation based on LDA," *CEUR Workshop Proc.*, vol. 1318, pp. 42–48, 2014.
- [3] R. Farzan and P. Brusilovsky, "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)*, 2006, vol. 4018 LNCS, pp. 91–100.
- [4] F. Le Roux, E. Ranjeet, V. Ghai, Y. Gao, and J. Lu, "A Course Recommender System Using Multiple Criteria Decision Making Method," *International Conference on Intelligent Systems and Knowledge Engineering*, no. 1, pp. 346–350, 2007.
- [5] J. Cuseo, "Academic Advisement and Student Retention: Empirical Connections & Systematic Interventions," *Natl. Acad. Advis. Assoc.*, p. 25, 2002.
- [6] C. Y. Huang, R. C. Chen, and L. S. Chen, "Course-recommendation system based on ontology," *Proc. - Int. Conf. Mach. Learn. Cybern.*, vol. 3, pp. 1168–1173, 2013.
- [7] Z. K. Zhang, T. Zhou, and Y. C. Zhang, "Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs," *Phys. A Stat. Mech. its Appl.*, vol. 389, no. 1, pp. 179–186, 2010.
- [8] I. Garcia, L. Sebastia, and E. Onaindia, "On the design of individual and group recommender systems for tourism," *Expert Syst. Appl.*, vol. 38, pp. 7683–7692, 2011.
- [9] D. Jannach, M. Zanker, A. Felfering, and G. Friedrich, "Recommender Systems An Introduction," *Teach. RS*, pp. 1–335, 2011.
- [10] R. Ren, L. Zhang, L. Cui, B. Deng, and Y. Shi, "Personalized Financial News Recommendation Algorithm Based on Ontology," *Procedia - Procedia Comput. Sci.*, vol. 55, pp. 843–851, 2015.
- [11] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. February, pp. 76–80, 2003.
- [12] W. Y. W. Yang, Z. W. Z. Wang, and M. Y. M. You, "An improved collaborative filtering method for recommendations' generation," *2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)*, vol. 5, pp. 4135–4139, 2004.
- [13] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," 2001.
- [14] Y. Lee, "Recommendation System Using Collaborative Filtering," p. 49, 2015.
- [15] X. Jin and B. Mobasher, "Using semantic similarity to enhance item-based collaborative filtering," *Proc. 2nd IASTED Int. Conf. Inf. Knowl. Shar.*, no. May, pp. 1–6, 2003.
- [16] R. Ljung *et al.*, "Content-Based Recommendation Systems," in *The Adaptive Web*, vol. 9, no. 4, 2010, pp. 325–341.
- [17] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artif. Intell. Rev.*, pp. 1–28, 2017.
- [18] P. C. Chang, C. H. Lin, and M. H. Chen, "A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems," *Algorithms*, vol. 9, no. 3, 2016.
- [19] H. Zhang, H. Yang, T. Huang, and G. Zhan, "DBNCF: Personalized courses recommendation system based on DBN in MOOC environment," *Proc. - 2017 Int. Symp. Educ. Technol. ISET 2017*, pp. 106–108, 2017.

- [20] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.
- [21] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.
- [22] S. Wang and K. Sapporo, "Topic-Oriented Query Expansion for Web Search Categories and Subject Descriptors," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 1029–1030.
- [23] L. R. Khan, "Ontology-based Information Selection," no. August, 2000.
- [24] The Universities and Colleges Admissions Service in United Kingdom, "UCAS."
- [25] F. Carballo, "Masters ' Courses Recommendation : Exploring Collaborative Filtering and Singular Value Decomposition with Student Profiling," no. November, 2014.
- [26] H. Zhang, T. Huang, Z. Lv, S. Y. Liu, and Z. Zhou, "MCRS: A course recommendation system for MOOCs," *Multimed. Tools Appl.*, vol. 68, no. 1cat2e, pp. 1–19, 2017.
- [27] N. Bendakir and E. Aimeur, "Using association rules for course recommendation," 2006.
- [28] M. M. Lotfy and a a Salama, "Subject Recommendation Using Ontology for Computer Science ACM Curricula," *Int. J. Inf. Sci. Intell. Syst.*, vol. 3, no. 1, pp. 199–205, 2014.
- [29] Hongji Yang, Zhan Cui, and P. O'Brien, "Extracting ontologies from legacy systems for understanding and re-engineering," *Proceedings. Twenty-Third Annu. Int. Comput. Softw. Appl. Conf. (Cat. No.99CB37032)*, pp. 21–26, 1999.
- [30] K. Zhou, S.-H. Yang, and H. Zha, *Functional matrix factorizations for cold-start recommendation*. 2011.
- [31] Y. Lee, "An Intelligent Course Recommendation System," *Smart Comput. Rev.*, vol. 1, no. 1, pp. 69–84, 2011.
- [32] C. Obeid, I. Lahoud, H. El Khoury, and P.-A. Champin, "Ontology-based Recommender System in Higher Education," *Companion Web Conf. 2018 Web Conf. 2018*, vol. 2, pp. 1031–1034, 2018.
- [33] X. Zhang, "Civil engineering professional courses collaborative recommendation system based on network," *2009 1st Int. Conf. Inf. Sci. Eng. ICISE 2009*, pp. 3253–3256, 2009.
- [34] H. Imran, M. Belghis-Zadeh, T.-W. Chang, Kinshuk, and S. Graf, "PLORS: a personalized learning object recommender system," *Vietnam J. Comput. Sci.*, vol. 3, no. 1, pp. 3–13, 2016.
- [35] M. H. Hsu, "A personalized English learning recommender system for ESL students," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 683–688, 2008.
- [36] H. Kadima and M. Malek, "Toward ontology-based personalization of a recommender system in social network," *Proc. 2010 Int. Conf. Soft Comput. Pattern Recognition, SoCPaR 2010*, vol. 5, pp. 119–122, 2010.
- [37] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, "Context-aware intelligent recommendation system for tourism," *2013 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2013*, no. March, pp. 328–331, 2013.
- [38] N. Y. Asabere, "Towards a viewpoint of context-aware recommender systems (CARS) and services," *Int. J. Comput. Sci. Telecommun.*, vol. 4, no. 1, pp. 10–29, 2013.
- [39] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.
- [40] J. Sandvig and R. Burke, "Aacorn: A CBR recommender for academic advising," *Tech. Rep. TR05-015*, 2005.
- [41] Y. Huang, "A Hybrid Recommender: Study and implementation of course selection recommender engine," 2017.
- [42] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.
- [43] N. Guarino, D. Oberle, and S. Staab, "Handbook on Ontologies," 2009.
- [44] G. Antoniou, F. Van Harmelen, and S. Edition, "A semantic Web primer," *Choice Rev. Online*, vol. 46, no. 03, pp. 46-1523-46-1523, 2008.
- [45] M. Grazia, D. Bono, G. Pieri, and O. Salvetti, "WP9: A Review of Data and Metadata Standards and Techniques for Representation of Multimedia Content," 2004.
- [46] D. Werner, C. Cruz, and C. Nicolle, "Ontology-based Recommender System of Economic Articles.," *Webist*, pp. 725–728, 2013.
- [47] Q. Yang, J. Sun, J. Wang, and Z. Jin, "Semantic web-based personalized recommendation system of courses knowledge research," *Proc. - 2010 Int. Conf. Intell. Comput. Cogn. Informatics, ICICCI 2010*, pp. 214–217, 2010.
- [48] R. Ambikapathy, "Ontology Based E-Learning System for a University on Semantic Web," *Int. J. Math. Comput. Appl. Res.*, vol. 1, no. 2, pp. 18–27, 2011.
- [49] K. Bagherifard, M. Rahmani, M. Nilashi, and V. Rafe, "Performance improvement for recommender systems using ontology," *Telematics and Informatics*, vol. 34, pp. 20

1772–1792, 2017.

- [50] P. Singto and A. Mingkhwan, “Semantic Searching IT Careers Concepts Based on Ontology,” *J. Adv. Manag. Sci.*, vol. 1, no. 1, pp. 102–106, 2013.
- [51] N. F. Noy and D. L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology,” *Stanford Knowl. Syst. Lab.*, p. 25, 2001.
- [52] S. V. Jambhulkar and S. J. Karale, “Semantic web application generation using Protogé; tool,” in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, 2016, pp. 1–5.
- [53] K. Lang, “NewsWeeder: Learning to Filter Netnews,” *Proc. 12th Int. Mach. Learn. Conf.*, pp. 331–339, 1995.
- [54] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.
- [55] G. Shani and A. Gunawardana, “Evaluating Recommendation Systems,” in *Recommender Systems Handbook*, 2011, pp. 257–297.
- [56] J. Beel and S. Langer, “A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9316, no. x, pp. 153–168, 2015.
- [57] F. Hernández del Olmo and E. Gaudioso, “Evaluation of recommender systems: A new approach,” *Expert Syst. Appl.*, vol. 35, pp. 790–804, 2008.
- [58] I. Brace, *Questionnaire design: how to plan, structure and write survey material for effective market research*, vol. 42, no. 06. Kogan Page, 2005.
- [59] L. Zhang, “The definition of novelty in recommendation system,” *J. Eng. Sci. Technol. Rev.*, vol. 6, no. 3, pp. 141–145, 2013.
- [60] A. Parameswaran, P. Venetis, and H. Garcia-Molina, “Recommendation Systems with Complex Constraints: A Course Recommendation Perspective,” *ACM Trans. Inf. Syst.*, vol. 29, p. 33, 2011.