4. PERSONALIZED COURSE RECOMMENDER SYSTEM

4.1 INTRODUCTION

This research work reviewed various Recommendation Systems (RS’s) and information retrieval systems. Based on the literature survey, Hybrid Recommendation System (HRS) architecture has been proposed. In this chapter, the overall working of the HRS is discussed. The key feature of the Recommender Systems is to support course personalization on the basis of learner’s domain of interest and domain. RS is considered an exceptional tool that helps in a non-formal learning environment by catering learner’s needs. The advantage of such a paradigm is that it helps to promote active learning, motivates learner and helps in enhancing their performance level. The hybrid recommender system will help in streamlining the course selection process and thus avoids time-consuming and brainstorming session for both faculty and scholars. One of the important aspects of the framework is that different information retrieval, machine learning, and data mining approaches were employed to take advantage of each individual technique. Thus proposed RS not only optimizes the prediction quality of recommendations but also resolve the bottlenecks of conventional individual techniques.

The pre-processing and post-processing stage of learner’s information and input query (research titles) are explained in this chapter. The user’s information is stored, and the query is passed to the RS where query is tokenized first, and then expansion technique will find a synonym for the tokenized words. The original query and the expanded query is then converted into N list of categories to make Sem-grams for maximizing the information retrieval to provide better recommendations. The recommendation module suggests courses to the learners based on N-Gram, query expansion and ontology based approach. In the end, the results are arranged using term frequency and are exposed to the user in the user display module. The course ontology has a key role to play in knowledge modelling and helps to give user results in respective domains. Thus with all requirements, the proposed HRS enclose the functions within a system.
4.2 PROPOSED RECOMMENDER SYSTEM METHODOLOGY

The principal objective of the research work is to develop a framework for Recommendation System with the help of better information retrieval techniques, ontology and using Threshold Based Nearest Neighbourhood (TBNN) approach. The system will be based on N-Gram, Query Expansion methods and TBNN filtering principles. The recommender system tries to suggest and personalizing courses for each scholar concerning their domain of interests and requirements.

It has been discussed in literature survey or chapter 2 that hybrid approach contributed most to the success of the new recommender systems as Hybrid filtering will benefit from the advantages of all individual methods incorporated by mitigating their weakness. The general process of Recommender System is shown in Figure 4.1

![Figure 4.1 General Process of Recommender System](image)

Personalization is becoming an important feature due to the differences among learners. The Recommender System is widely accepted as solutions to overcome the personalization issue and information retrieval challenges [94]. It was suggested that hybridization of Individual information retrieval technique with other recommender techniques could improve the efficacy of e-learning system. New e-learning systems should come out to make sure the personalization of resources not only in terms of content but also concerning the courses.
In non-formal educational, it is quite tricky for a learner to differentiate the courses that would be important for a domain if learners are not expertise in it. To fine-tune the process of course selection in e-learning systems and non-formal learning, the recommender systems are advocated to dispense information related to courses and facilitate the learner as well as an instructor to acquire knowledge about the courses related the domain of interest. The proposed hybrid recommender systems flow through the various phases like creating Domain Model, creating Learner Profile, Query Pre-Processing, Query expansion, Filtering using TF-IDF, Similar Learner Generation and Technology Acceptance Model to check the feasibility and effectiveness of Recommender System.

4.3 OBJECTIVES OF PERSONALIZED SYSTEM

In proposed work, the new approach effectively makes the hybrid recommendation system for e-learning systems to successfully tackle the constraints of personalization and the current recommendation techniques. The primary aim behind the current research work is dedicated to design a suitable RS architecture for course personalization. The key objectives of the recent analysis are detailed as follows.

- The introduction of the recommender system framework with the support of the hybrid techniques and ontology
- A huge number of collection and storing of learning courses
- Knowledge modeling of courses as per domain to support personalization and a better understanding of domain courses
- The retrieval and delivery of the most relevant courses from storage space in a more efficient way and to suggest them by learner’s concerned area
- To alleviate the performance, interest and knowledge based skills of the learner
4.4 PROPOSED SYSTEM ARCHITECTURE

In general, the basis is to have the paradigm of modular and lightweight architecture that can be incorporated in an e-learning system to support personalization of resources for a new learner.

Figure 4.2 Architecture of the Proposed System

The general architecture of the proposed Hybrid Recommender system is above shown in Figure 4.2. The personalized recommender system represents a model of conceptualization and elucidating the user who may come from different background using the personalized ontology of learner's model. The innovative idea of personalizing courses for a learner is generated from the research title
supplied as a query which is supported by ontology and the technique developed for personalization. At the outset, the learner profile is created, and the background, knowledge, expertise level and experience related data are gathered in a local repository as knowledge base for future use. The repository at the end includes the information collected from the user and the personalized courses suggested by the system. So, the knowledge base stored in a local repository can be used to generate similar learners for a new user having similar background and choices [95].

Learner query is treated in a way so that it will help to retrieve more useful informational which is in the interest of the learner and the disambiguation of the query is done by using query expansion technique. In this research work, the query disambiguation has been applied to search synonymy for the user query words by the use of global resources in the form of Wordnet. The use of the equivalent word mapping for each query term can increase the efficacy of the recommendation process. The course ontology is created by means Web Ontology Language (OWL) with the help of Protege in an innovative approach. The step of mining related courses from the developed ontology is elegantly done to provide more choices in the form of closely related to the learners.

Finally, the courses extracted were suggested to the user to get some knowledge and insight about the query given as input. Consequently, considering the recommendations, both the learner and the guide is proficient enough to gain a fair understanding of the learner requirement on a specific topic. In the end, the learner can choose, or guide can offer the courses according to the requirements of the learner based on the learner’s knowledge, background and domain of interest. The use of the ontology rules was explored in this context. Here, the main importance is given to the development of course ontology and the relationship among courses according to the major domains of computer science.

4.5 SEQUENTIAL PROCESS OF RECOMMENDER SYSTEM

The initial process begins with the development of the course ontology, followed by the two sequential processes explained below [96].
**Sequential Process 1**

**Step 1:** The user information is gathered with the help of a registration form and the login is created for a particular user.

**Step 2:** After login, the input query provided by the user is given to the Recommendation generator for further processing. The user query can be of any length usually the research titles.

**Step 3:** The space tokenizer is used to tokenize the query into separate keywords known as tokens.

**Step 4:** The tokenized keywords are sent for the removal of the words that are of least importance in a query called stop words.

**Step 5:** After stop word removal the rest of the tokenized words are sent for further processing of query expansion.

**Step 6:** The query expansion processor searches the synonyms for each tokenized query terms in the Wordnet for the equivalent term.

**Step 7:** The original terms whose synset are not available in WordNet are kept as such and the new separate query is formed from synset and the words whose synset are not generated.

**Step 8:** The tokens and the synset generated from Wordnet are sent to recommendation generation module where they are processed by N-gram generator to convert the words into bi-grams and tri-grams.

**Step 9:** The newly generated query list then retrieves the courses matching with these grams are sent to another module for further processing.
Step 10: The redundancy is removed, and the results are re-ranked using the term frequency algorithm. For all unique courses the related courses are retrieved from the course ontology to give user more options to choose courses of their interest.

**Sequential Process 2**

Step 1: The user information gathered over the time will be used as a knowledge base to generate neighbours (similar learners) for a new targeted learner.

Step 2: The Learners are classified based on their domain of interest and the background knowledge.

Step 3: For every case (tuple) that is set to be predicted from the target data set, K closest members are searched for a new learner using the Euclidean function on the parameters mentioned in step 2. The value of K is the number of nearest neighbours to retrieve. The main function is to relate the unknown to the known according to some distance / similarity function.

Step 4: To find out nearest neighbours sort the distances calculated at step3 based on the Kth minimum distance.

Step 5: Calculated threshold for all nearest neighbors as the best prediction value of new nearest neighbor or similar learner.

### 4.6 FRAMEWORK OF PROPOSED RECOMMENDER SYSTEM

The comprehensive workflow of each major component is explained below. The innovative proposed framework encompasses the following important stages.
4.6.1 Course Modelling

In this study, an architectural methodology was presented in chapter 3 for introducing and maintaining ontology based knowledge modeling with a focus on the knowledge in the domain Computer Science (CS). This study is concerned about the product centred view of knowledge scattered in CS and mainly focused on the documenting the knowledge, its creation, storage and reuse in another application. The idea of course modelling is to explicate, formalize and documenting the domain knowledge to provide it as a sort of tangible resource, and a proposal to support the knowledge development for individual learner’s by presenting the right information sources in an appropriate form.

Figure. 4.3. Knowledge Management Approaches
Figure 4.3 explains the distinctive software support for different methods to knowledge management. This transition from implicit (Intangible) to explicit (Tangible) knowledge modelling in the form of certain standards not only allows enhancing the structural capacity of a domain but also increases flexibility and creativity [97].

The primary task of ontology development is to get the concepts of a domain in a proper hierarchy together with the concept of relationships and properties. The illustration of the ontology will demonstrate the relationship between the courses in a domain and for that purpose OWL format being used by protege to represent the ontologies. Some of the major domain in the CS field [98] is shown in Figure 4.4.

![Conceptual Hierarchy of CS Domain](image)

**Figure 4.4. Conceptual Hierarchy of CS Domain**

Course ontology constitutes one of the core components of the proposed recommender system which will help to provide the learner with a better way of understanding the domain of interest. The domain expert will be responsible for creating relationships between courses that are available in the databases along with ontology rules that will be applicable to them. Once the task of creating a
relationship is over, the same set of ontology rules can be applied to similar domains and it should be remembered that relations can be increased or decreased also. The knowledge model in the form of domain ontology as shown in Table 4.1 will be helpful for easily tracking the concept present in a domain by manoeuvring through the domain ontology which rather gave more course prudence recommended by the system.

Table 4.1 OntoGraf generation for domain ontology
Ontology based application are the basic elements for semantic web because its primary focus is on the domain concept and relationship, and Protégé is currently the most popular tool for the development of domain ontology. The responsibility of domain expert is to create ontology and relationships among the concepts which are basic criterion for developing ontology for a domain. Its nature should be flexible while all relationships should be defined in advance and it is clear that the course model generated provides a better understanding of the domain.

4.6.2 Learner Modeling

The module intends to collect the data about the learner during registration process to create the learner’s profile. It includes the attributes like personal information, background, and previous knowledge, the domain of interest, expertise level and performance. After submitting all the information, the learner will be made to log in to the system to proceed further into the recommendation system. The learner will be made to enter the title of interest in the form of a query to generate recommendations. The information provided by the learner is then used in creating learner model, in the form of ontology as shown in Figure 4.5 and Figure 4.6 in the shape of OntoGraf. The things that are associated with the learner model are presented and can be easily understood using graphical structure.

Figure 4.5 Learner’s OntoGraf
The information provided by the learner is stored in Learner model e.g. personal information, prior knowledge (degree, course), and domain of Interest (networks, dbms, software engineering, cloud and data mining etc.). All information is collected during the process of learner registration, and stored in the database in a way so that the information later can be viewed in the form of a graph and also used for neighbourhood generation.

Currently, the ontology development is mostly done from scratch, because of several reasons. First, they are modified to work for particular applications that will restrict its reusability and potential. Second, developers usually follow a single tier approach at the time of developing ontologies, those typically cover different domains, thus hinder the reusability of appropriate parts of ontology for other applications. Third, it is difficult to come across an appropriate ontology because of lack of proper standards to document the ontologies and its supporting tools.

Knowledge sharing and reuse are one of the main motivating factors for semantic web driven applications. The ontologies are being seen as an alternative to create some common understanding to mitigate the semantic heterogeneity. The ontology in different domains of application has evinced its mettle in different ways. Even though, less effort was made to develop ontology for information systems domain as compared to other domains. Ontology
developed in present study will provide as a knowledge representation model for modern educational systems.

4.6.3 Learners Query Processing

- **Tokenization**

  Tokenization is the procedure where a sentence is broken at white spaces into separate words as keywords, phrases, and symbols called tokens. The learner’s query is given to the tokenizer available in recommendation generation module. The tokenizer divides text available in the form of the query into separate word structure called as tokens. A token is nothing but a contiguous string of characters in a word normally done after stemming process in various studies. The tokenizer component is shown in Figure 4.7

![Figure 4.7 Tokenizer Components in Recommender System](image)

Depend on the usage tokens can be a sentence, phrase or an individual word and are normally separated by the characters, like whitespace or a line break, or by some punctuation character. The process in Figure 4.8 explains the working of a tokenizer on a learners query. In this study, tokenization is performed by advanced conventional splitting method over the spaces of the query terms. The tokenization is a key phase in pre-processing stage as it is useful in dividing the query sentence into a word list for further processing.
Stemming is, in fact, a normalization process in linguistic, which removes the inflexional endings from words or reduces alternative form of English words to a standard form by restricting the space of vocabulary. Stemming relates the words of different forms based on the same stem and connects them with a root word. Mostly it is used for the information retrieval process for mapping the words with a common root which proves useful in retrieving more relevant information. The strict and fast thesaurus based example of stemmer is porter stemmer and in this study, it was optional choices since n-gram exclude this process. In Figure 4.9 a sample input query text segmentation process is shown.

**Example**

<table>
<thead>
<tr>
<th>Input Query</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biometrics Security and Forensics in Digital Passport System</td>
<td>Biometrics, Security, and, Forensics, in, Digital, Passport, System</td>
</tr>
<tr>
<td>Knowledge Discovery Process on Databases Using Artificial Intelligence</td>
<td>Knowledge, Discovery, Databases, Artificial, Intelligence</td>
</tr>
</tbody>
</table>

**Figure 4.8 Tokenizing Process**

**Figure 4.9 Input Query Tokenization**
• **Stop Word Elimination**

The input query from the learner is analyzed and assessed by the exclusion of the stop words. The process of stop word removal is initiated to lessen the noisy or least important words of the input query. After that, the remaining query words are organized for facilitating its lucidity and continuity. In this regard, the query pre-processing is the first process, which is well geared to eliminate the non essential query words effectively.

The words which are virtually insignificant are usually labelled as the stop words, such words include articles, conjunctions, and the prepositions which are eliminated as they do not possess any type of informatics. In this study, some additional words which are of less importance or logically effect recommendation results are also included in the removal list. The stop word removal has a significant advantage in the next processing stage of recommender system as it limits the generation of Sem-grams. In Figure 4.10 the example of the query with tokenized and stop word removal form.

![Figure 4.10 Stop Word Removal Process](image)

The approach used to design the stop-word removal algorithm and its implementation uses a basic dictionary based approach. In this approach, a predefined list of words which needed to be removed is compared to the target query on which removal is required. The sequential steps for stop word removal process are given below, and the examples are given in Figure 4.11.
Stop Word Removal Process

Step 1: The input query is converted into tokens which are stored as a list.

Step 2: The stop words are read from stop word list file and compared with each token using sequential search technique.

Step 3: Matched words in the token list file are separated, and the comparison continues till the token list exhausted.

Step 5: Repeat step 2 and step 3 until stopword file is compared.

Step 6: The spared tokens in the list send for the next stage of processing.

Example

<table>
<thead>
<tr>
<th>Input Query</th>
<th>Content based image Retrieval techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query after Stop word removal</td>
<td>Content, image</td>
</tr>
<tr>
<td>Input Query</td>
<td>Biometrics Security &amp; Forensics in digital passport system</td>
</tr>
<tr>
<td>Query after Stop word removal</td>
<td>Biometrics, Security, Forensics, Digital</td>
</tr>
</tbody>
</table>

Figure 4.11. Stop Word Removal Examples

4.6.4 Query Expansion

In a real-world scenario, mostly the query provided by the users contains item terms that do not exactly match with most of the relevant items, and this mismatch of words makes retrieval of information more complicated and less concerned. Even important items sometimes are named with dissimilar terms in the database unlike those present in the query. To resolve such issues, it is indispensable to modify user query words, and as a consequence query
expansion technique was proposed to overcome these problems. Query expansion mechanism starts by reformulating the active query or creating a new query by further adding more significant words into the base query for improving information retrieval effectiveness. Many expansion methods have been suggested, but Query expansion as can be seen in Figure 4.12 remains one of the most effective among all other approaches for improving the performance for ad-hoc retrieval.

![Figure 4.12. Query Expansion Techniques](image)

4.6.4.1 Query Expansion Issues

The most significant and the basic problem with information retrieval is a disparity among the query words in a way that if a user will use some words in query and the author might have labelled the same concept with different synonyms. It would be very difficult in such a case to retrieve the items that will match the query. Various methods were introduced for expansion of the user query and some of the methods are interactive, manual and automatic in nature. But sometimes a user is unable to give information which is enough to expand the query or a user cannot judge the most suitable synonym for the query words. In such scenario automatic query expansion is required which do not require user participation and if the users are unsatisfied by the recommendations generated, fresh recommendations will be generated with modified words that can suit the learner requirements. Query expansion not only satisfying the requirements related to learners but also play a significant part to increase the efficiency of the recommender system without much of the learner involvement.
The automatic query expansion using WordNet was among the approaches that has been the intention of this research because even if one relevant course will be retrieved using query expansion technique, which means a 20% increment in the recall. In this global study method of expansion is used for new query reformulation which modifies or creates a new query based on global resources and the main information used is synonymy. A few regular query expansion properties and types are given below.

- **Global methods**: Some of the properties of these methods are as follows.
  
  - They work autonomously of the input query and result set
  - They start finding word relationships in their corpus
  - They use external Thesaurus and Wordnet as external source for synonyms and spelling check

- **Local methods**: The property of such a method is
  
  - They analyze the result retrieved by primary query search

The query expansion is the right candidate that helps to increase the true positive rate or recall. An example of the word and their synonyms is shown in Table 4.2.

<table>
<thead>
<tr>
<th>Word</th>
<th>Nearest Neighbour Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing</td>
<td>artificial intelligence, AI, applied science</td>
</tr>
<tr>
<td>Picture</td>
<td>image, icon, graphic, art, scene, movie, film, visualize</td>
</tr>
<tr>
<td>View</td>
<td>opinion, sentiment, aspect, scene.</td>
</tr>
<tr>
<td>Monitor</td>
<td>supervise, observation, surveillance.</td>
</tr>
<tr>
<td>Investigation</td>
<td>analysis, examination, scrutiny, examination, testing, inquiry</td>
</tr>
</tbody>
</table>
4.6.4.2 WordNet Query Expansion

The multilingual lexical database popularly known as Wordnet was created by a group at Princeton institute lead by Miller. The database is able to serve both as online thesaurus and an online dictionary and the researchers are using the same since its commencement. Few applications have used the WordNet such as Information systems and retrieval, word sense disambiguation, text summarization and classification have used the WordNet. The synonym set or synset are the main elements of WordNet which separate the lexicon dictionary into nouns, adjectives, verbs and are organised in the form of synonyms. The synonym or synset represent the concept that all the words contained in it can be interchanged. The knowledge a WordNet incorporate consists of a word definition and a pointer which binds the two words together in syntax. The primary intentions of WordNet are to help its users in automatic analysis of text and provide a permuted combination of both dictionary and thesaurus. In present research work the WordNet database were used to generate the synset shown in Figure 4.13 which is helpful in generating the more relevant recommendations in a case when a recommender system receive very less information from the learner queries.

Figure 4.13 Synonyms Retrieved Related to the Query Words
4.6.5 Sem-gram Conversion

In Sem-gram approach the original tokens and their semantic (synonym) tokens are used to generate a list of queries using N-gram technique. N-gram is a sequence of co-occurring words from a given sequence of text. The N-gram texts are extensively used in text mining and natural language processing for feature extraction in supervised learning. In this study, N-gram has been used with query expansion techniques to generate recommendations and presents an evaluation method for a given user query with respect to specific word order. The base of the evaluation method is to reorder the words to create n-grams for maximizing the relevant information retrieval for user recommendation. The key parameter of the proposed methodology incorporates unigrams, bigram and trigram probability which corresponds to the frequency of the relevant information retrieved from the dataset. The relative advantage of n-gram methodology is that it goes well with a big set of words and repudiates the arduous and expensive procedure of creating error patterns by simply collecting word order errors.

N-gram is a prediction technique that uses probabilistic methods for predicting the next character or the word after observing N-1 words, and while, computing N-grams there is a movement of one word forward. Thus, the probability of computing the next word is very much allied to compute the probability of a sequence of words. N-Gram extracts the tokenized words from the input query and runs each word all the way through the N-Gram filter as they are word sets within a specified window. The foremost intention to use N-gram in this study is to classify a given query Xi into a list of N number of categories Xi (X1, X2, X3, and X4 ... Xn) to increase the keyword generation which may maximize the information retrieval.

If X=Num of words in an input query sentence K, the number of N-grams for sentence K would be as in equation 4.1.

\[ \text{N-grams K} = X - (N-1) \quad \text{Eq. (4.1)} \]
With \( N = 1 \) the Model is called Unigram, \( N=2 \) Model is Bigram and \( N=3 \) is Model is Trigram. Bi-grams are also known as Markov assumptions, which assume that the probability of the upcoming word can be predicted by checking at the last word encountered. To generalize bi-gram to tri-gram, two last words in the past needs to be checked, similarly for \( N \)-gram by checking the past \( N-1 \) words. Hence, for the probability of next word in a sequence would be given by the general equation 4.2.

\[
P (W_n | W_1^{n-1}) \approx P (W_n | W_{n-N+1}^{n-1}) \quad \text{Eq. (4.2)}
\]

Where the sequence of words \( W_1, W_2... W_{n-1} \) is represented as \( W_1^{n-1} \).

\( N \)-grams are able to provide an estimation of the probability of observed word sequence \( W \) of \( P (W) \), and assume that the probability of a given the word in a sentence depends on the fixed number of previous words, the probability of \( N \)-word string can be written as provided by equation 4.3.

\[
P ( W_n | W_1 ...W_{n-1}) \quad \text{Eq. (4.3)}
\]

The probabilities of the gram generation are as follows

- **Unigrams:** \( P (W_1 W_2...W_n) = P (W_1) P (W_2).....P (W_n) \)
- **Bigrams:** \( P (W_1 W_2...W_n) = P (W_1) P (W_2|W_1)....P (W_n|W_{n-1}) \)
- **Trigrams:** \( P (W_1 W_2...W_n) = P (W_1) P (W_2|W_1).... P (W_n| W_{n-2} W_{n-1}) \)

The best and easiest way to estimate the probability of a gram is to use Maximum likelihood estimation, based on normalizing the count in the interval of \([0, 1]\) by taking the count from the corpus. For example to compute the probability (see equation 4.4) of a bigram word \( B \) followed by \( A \) is to count the bigrams \( C \) (AB) from the document and normalize it with the number of bigrams that starts with \( A \).
C (W_{n-1}) in the denominator is the count of bigrams starting with W_{n-1}, and the bigrams starting with W_{n-1} is equal to the number that W_{n-1} occurs in Corpus.

Keeping in view that a query sentence may include the accurate words but their order is not right. It is possible that by generating its permuted sentences or words reordering one of them might be the correct sentence (words in right order). Hence, N-grams are the best approach to repair the word order in a new query list. N-gram also reorders the words in bi-grams trigrams and n-grams sentence so that number of hits should maximize to improve recommendation. Table 4.3 shows the use of a confusion matrix for generating the grams for an input sentence as.

\[ S = [W[1], W[2]...W[N-1], W[N]] \quad \text{Eq. (4.5)} \]

\[
P(W_n \mid W_{n-1}) = \frac{C(W_{n-1} W_n)}{C(W_{n-1})} \quad \text{Eq. (4.4)}
\]

Table 4.3. Confusion matrix construction of order N x N

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Word[1]</td>
<td>P[1,1]</td>
<td>P[1,2]</td>
<td>P[1,3]</td>
<td>......</td>
<td>P[1,N]</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>

The length of the sentence determines the size of the confusion matrix. The objective of confusion matrix is to extort the valid bigrams according to the language model. Unlike other approaches, the n-gram method can be applied to
any query in any other language that does not work with a particular word set. Even the use of tagger and parser is not essential and it does not require the manual collection of words.

All searching techniques have a different approach, and in current study, the methodology used is left to right searching. Permutation filtering searches for bigram and trigram from a unigram input sentence. N-gram is best and straightforward way to reconstruct a query with the words error order to reorder them by using all the possible permutations. In the past several techniques were developed to solve the problem of reordering using the parser and other rules but the success was very low. The only drawback for the permuted approach was that there would be N! Permutation for a sentence with a length of N words and this number seems large to restrict the further processing of query words. But the problem is mitigated with the help of stopword removal technique which allows removing the least essential words from the query sentence and thereby reducing the length of sentence and N! Permutations. In Figure 4.14 unigrams, bigrams and trigrams generated can be seen for a user query.

![Figure 4.14 N-gram Generation for an Input Query](image-url)
4.6.6 Course Information Extraction

The courses that are retrieved from the database as a result of n-gram processing may contain redundant data in the form of courses itself. To remove redundancy and display courses as per ranking Term Frequency is used for this purpose. Term frequency is a technique used in text mining for document categorization, in information retrieval and sentiment analysis even if the goals of both the techniques are distinct. TF classify the documents while as sentiment analysis does opinion classification into positive and negative. This shows that even if algorithms are similar, the result of each gives a distinctive insight. In this study, Term Frequency is used for courses categorization and additionally to remove course redundancy. One more advantage of term frequency is that it needs not to be trained well ahead of time as it will automatically check the differences in the document. TF is a measurement in statistics that is used to assess the magnitude of a word in a corpus. The significance of a word increases proportionally to the number of times that word appears. Different variations of the Term Frequency- Inverse Document Frequency (TF-IDF) methods are used by search engines and as a central to check the score and rank the document's relevance provided by the user query. Term frequency is one of the simplest ranking techniques that can be easily used to filter in various domain fields including courses.

Term frequency, in general contains two terms in which the first computes the normalized value of Term Frequency (TF), and the second term computes the Inverse Document Frequency (IDF), as the log value of the number of the documents divided by the number of documents where a particular term appears. Since this study deals with term frequency only which is calculated as follows.

\[
TF = \frac{\text{The number of times a course appears in a retrieved list}}{\text{Total number of Courses}} \quad \text{Eq. (4.6)}
\]
Term Frequency, basically measures how frequently a course occurs in a retrieved list of information. Since every course is different in length, it is possible that a course would appear multiple times because of the query list generated by n-gram technique. The courses retrieved after TF filtering are then being searched for their related courses from the course ontology file with the help of a relation Is-related (relationship property created between the two courses) from the course ontology file. Both the results are then being displayed to the learners so that user can have maximum choices of courses to be selected for their study program.

### 4.6.7 Similar Learner generation

In the proposed Recommender System it is believed that stored learners information will serve as a knowledge base and may help to find a similar learner for the user who is new to the system. As discussed in chapter 1 the course selection process is influenced by the social factors also. Thus all learners which are having some characteristic similarity among themselves form a good neighbourhood for the new learner. The recommendation will be filtered for a new learner using the learner’s data set and the learners whose similarity measure will fulfil threshold criteria will be treated as best among the recommendations. To find neighbours an algorithm for checking similarity between the prior learners’ and the newly targeted learner based on Euclidian distance measures is used.

This approach does not demand neighbours as input instead by measuring the distance to place two learners together with similar in characteristics. The similarity measure is used for the textual data classification for KNN and might give a different result based on value of K [99]. The KNN algorithm use Euclidean distance measure and the threshold value to find a learner who will be similar as that of a targeted learner. The tuple in the dataset which does not meet up the criteria of threshold value will not be considered for recommendation rather they will be discarded even if they have minimum distance value. This indicates that only those learners which meet the threshold value are send as recommended neighbours. The experimental results proves that TBNN filtering provide more appropriate and quick results. As a result the system that has been proposed will
not only solve the rating problem but overspecialization and the curse of dimensionality problem of individual recommender systems.

The KNN in this study was chosen because for following reasons.

- **Instance-based Learning**: K-Nearest Neighborhood approach belongs to the supervised learning category which is an instance based method used for retrieving the neighbourhood content

- **Lazy Learning**: In K-Nearest Neighborhood the work is done when a request for prediction is received. The approach do not require learning model and that is the reason it is often referred as Lazy Learning Algorithm

- **Non-Parametric**: K-Nearest Neighborhood does not make any speculations about the practical form of the issue being solved. The K-Nearest Neighborhood also does not use any parameter and is mostly used for regression and classification problems.

Most of the conventional approaches consider rating feature during recommendation while as in proposed system this rating attribute is not considered rather certain other learner features such as prior knowledge, domain preference, study background, performance and expert level are considered to generate the recommendation in the form of similar learners or nearest neighbour

- The count of neighbours of a new learner should not be predefined rather it should be flexible

- The data points that are very much adjacent to each other should only be included for neighbourhood formation

- For a new learner the best neighbour would be the learner from the dataset which surpass a threshold value as given in below Figure 4.15. So, it is clear from the above discussed requirements that in proposed technique the art of finding alike learner is fairly dissimilar from a conventional approach.
The TBNN approach uses a Euclidean distance measure for placing together the learners who are having certain similar features and hold some threshold value. The algorithm that describe learners Li (i=1 to n) as a vector with some of the attributes like prior knowledge (about research like research project and Mphil, 0 depict learner do not possess any knowledge and 1 depict prior knowledge). Background study (MCA, M.Tech, M.sc, ME), Domain of interest, expertise level (0 means learner do not possess expertise level at all and 1 means the learner is an expert in its field) and performance (1 means good performance and 0 means no performance at all).

![Flow Chart for the Neighborhood Generation](image)

**Figure 4.15. Flow Chart for the Neighborhood Generation**
In current study, clustering has not been chosen because it allocates each learner to a group even if the learners are at much distance, which states that every time a cluster does not contain the adjacent learners. The aim of current study is to search the learners which are very close with respect to a particular learner. The neighbourhood technique is perfect without any complications mentioned above. To provide perfect recommendation by means of the neighbourhood technique, it is expected that learners which are alike and are using the experience of related learner are placed adjacent to each other.

The proposed algorithm calculates the similarity among the learners using above mentioned features and by Euclidean distance to compute the distance between new learner and the previous learners. Prior learners whose background and domain matches with the new learner should be considered as nearest neighbourhood and if it crosses a threshold value it is considered the best. The learner characteristics are normalized into values to ensure input for each of the characteristics. After normalizing the data Euclidian distance will be easy to compute between the learners on the basis of characteristics. The Euclidian distance is also known by L2 distance in machine learning applications, such as in K-Nearest Neighbour, K-Means Clustering. Euclidian distance is given in equation 4.7 where i and n is the number of characteristics of learners or data points and radius as a threshold value at later is used stage to verify best among the neighbourhood.

\[ \text{Euclidean Distance} = d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad \text{Eq. (4.7)} \]

With neighbourhood technique most of the limitations and challenges are mitigated all that recommender systems going through like Cold Start (at times it is very complex for a recommender system to suggest recommendations for a new user and a new item as the new item has not been rated yet by any user and the profile of the new users is not know because he has not rated any item usually in collaborative recommender systems). Overspecialization (when the recommended items are more alike, and the list is not different), Sparsity (which is nothing but lack of information about a particular item), Trust (this issue arises towards the evaluation of some particular type of users) etc.
4.6.8 Recommendation display

This is the Interface through which a learner will have interaction with the recommendation system while providing the input to the system, and the in-turn recommender system will send the courses and neighbours that are being recommended to a particular learner through this interface. The courses recommended can be selected, and the selection is stored against the particular learner in the database. The interface is used for both gathering information from the learner as well as the recommendation monitor where recommended courses are being suggested to a learner in precise, easy and in an Informative way. The recommendations will include the courses, their domain, course credits, department offering the course and similar learner’s details are also available. The user after choosing their final course list will leave the system and information will be reflected back into their profile.

4.7 CONCLUSION

In this chapter, a new hybrid approach based recommender system is explained in detail for suggesting courses. Unlike other existing individual recommender system, in this research work not only hybrid method of information retrieval were added to Personalized Course Recommender System (PCRS), but also ontology support was provided to extract more information for the recommendation. Instead of applying collaborative and content filtering based mining approaches, query classification and expansion based approach is used for query and information conversion. The learner input query is processed to convert the query into a list of N number of queries using the n-gram and synonym based techniques to retrieve maximum relevant information. The ontology plays a significant role in identifying the course information that is related to the course information extracted from hybrid approach. Also with the help of ontology the courses and their domain are easy to understand that is available in the domain ontology. Hence, to validate its use and impact prediction will be useful for enhancing the process of information retrieval.
The study proposes a hybrid recommendation system to overcome the limitation of the present conventional recommender systems knowing that hybrid technique will take the benefits of all individual techniques included because it will overcome the limitation possessed by the individual approaches. Thus appropriate courses and similar learners have to be suggested by checking the learner query and the learner’s knowledge base as the characteristics and interest are different for each learner. The underline work of this study has been the semantic nature of the courses and the generation of similar learner generation. The next step is to evaluate the recommender system with different parameters, so the following chapter will discuss the results obtained and the test bed used to assess the proposed system.

The proposed recommender system is useful, appropriate and was found more efficient for the learners when all the techniques were compared together experimentally. There are some valid reasons for implementing the recommender system and the prominent one is the satisfaction level of the learner’s and the second one is to enhance the monetary success of the proposal. The TBNN approach will give additional strength to the system by recommending the appropriate similar learners. The intent of searching the nearest neighbours is to find the learners who possess similarity with the targeted learner with respect to domain similarity and the learning preferences. The Future work will be to optimize the synset selection process automatically or through relevance feedback technique.