Semantic Search Engine using Joomla Framework with Modified tf-idf and TRApriori Algorithm

Yogendra Kumar Jain
S.A.T.I. vidisha (M.P)
India

Payal Saxena
S.A.T.I. vidisha (M.P)
India

Rustam Singh Rajput
S.A.T.I. vidisha (M.P)
India

ABSTRACT
As the amount of data available in a repository increases, content retrieval from the huge data stored in the repository becomes a tedious task. Though Content Management System helps us to manage the data, yet searching the relevant data is still a daunting task. For that, we need efficient Search Algorithms for maximizing the correlation between data required and data returned by semantic search engine. Many courseware repositories is an interface through which various students, teachers, etc can access on-line learning material, course contents, presentations, videos, lectures etc. in this paper we present a technique that automatically constructs ontology from the given courseware repositories. A search engine mechanism is developed that provide a semantic search capability based on a modified TF-IDF(term frequency inverse document frequency) weighting scheme and then determines the association among term through TRapriori algorithm. We evaluate our result with custom Google search engine.

Keywords:—Ontology; tf-idf; TRApriori algorithm;

1.INTRODUCTION

The phenomenal rise in the number of engineering colleges in India needs to be equally matched with efficient educational infrastructure and resources which in turn addresses the improvement in quality of teaching in the remote areas where there is a lack of efficient teachers. To make the quality of teaching at par throughout the country, Prof Deepak B. Phatak of IIT Bombay has taken an initiative called effective teaching learning of computer programming which is an online workshop to empower teachers in higher education, funded by the National Mission on Education through ICT[1] (MHRD, Government of India). To make the offerings accessible from every corner of the world, a web-portal is planned to be hosted which would contain the recorded video lectures, along with relevant study materials like slides, questionnaires and many more.

For developing these courseware most of the people are using different types of content management frameworks like Joomla, Drupal, Wordpressetc, We are using Joomla content management framework which is easy to understand and to apply as compare to other frameworks. But problem is that if contain is available large amounts of data in the form of videos, ppt, pdf, text then we will need to a semantic search engine for retrieve the optimal or appropriate result. The idea is to develop Semantic Search Engine with Joomla framework[2],xampp apache server[3] for National Portal for Professional Teachers".

2. RELATED WORK

This section describes different courseware repositories, a brief overview of domain ontology.

2.1 Courseware Repositories

There are various institutes across the globe which are willing to collaborate and participate in various endevours, which will help in raising their standard of technical teachings. But some hampered by of resources and some by geographical separations. Also, quality of teaching and study material is not uniform even in small geographical span. This portal will help in addressing these issue and will act as one of the important tool in bringing homogeneity across quality of course material and quality of teaching. Information pool available and its efficient use along with the guidance of experts of the field will help in achieving this goal. So we study three most popular courseware repository. OCW1, NPTEL2, CDEEP3. In OCW (courseware of MIT) There are over 2000 course in 36 different stream [4]. Most of documents are in pdf formats. In OCW If we search the key word like " operating system thread" it does not search the correct key word it search "Micro- kernels". It does not help the user in more advance topics. NPTEL (National Program for Technology Enhanced Learning) is the program that is conducted by MHRD (ministry of human resource department) of india. Which was started in 1999[5]. Its course repository In formats pdf, vedio and web documents and ppts. But there is no search option. CDEEP (Center for Distance Engineering Education Program) was started by IIT (Indian institute of technology) Bombay india. CDEEP is offing 53 course in 6 stream. It provide different services for distance education like class room lecture in pdf formats and audio and vedio formats and tutorial and different assignments . But there is no any searching option for particular topic.

[1]http://ocw.mit.edu
[2]http://nptel.iitm.ac.in
[3]http://www.cdeep.iitb.ac.in

2.2 Joomla Architecture

Joomla follows MVC architecture [6]. Model-View-Controller is a software design pattern that can be used to organize code in such a way that the business logic and data presentation are separate. The reason behind this approach is that if the business logic is grouped into one section, then the interface and user interaction can be revised and customized without changing the business logic. These three main roles are the basis for the Joomla MVC refers to the MVC architecture.

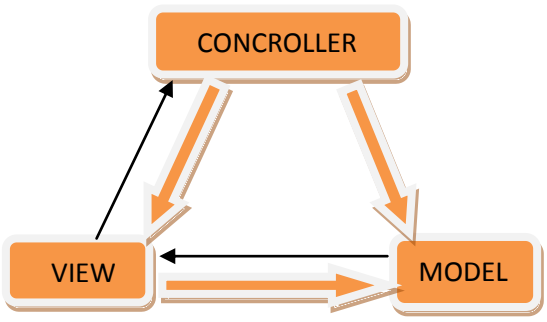

Fig. 1: Joomla MVC Architecture[6].

Model: The model will contain methods to add, remove and update information in the database. It will also contain methods to retrieve the data from the database. If developer wants to manipulate the data then the model is the only element that needs to be changed.

View: The view is the part of the component that is used to render the data from the model in a manner that is suitable for interaction. For a web-based application, the view would generally be an HTML page that is returned to the data. The view pulls data from the model and feeds the data into a template which is presented to the user. The view does not cause the data to be modified in any way, it only displays data retrieved from the model.

Controller: The controller is responsible for responding to user actions. In case of a web application, a user action is (generally) a page request. The controller will determine what request is being made by the user and respond appropriately by triggering the model to manipulate the data appropriately and passing the model into the view. The controller does not display the data in the model, it only triggers methods in the model which modify the data, and then pass the model into the view which displays the data.

All requests coming into the directory in which Joomla is located are redirected to the index.php le in that directory. The PHP les in the joomla cannot access directly by the user. If developer wants to create the PHP le he/she has to change the code in the component folder otherwise use the PHP component. It allows developers to create simple PHP pages and link them to the Joomla Menu. This makes developers to easily create a custom page without having to create a whole component.

2.3 Ontology

Ontology is a collection of concepts and their interrelationships, This system is a large number of ideas and concepts to gather in a hierarchical order. User also identifies the sense of terms to map those terms to concepts in Word Net. It provides a mechanism to capture information about the objects, Classes and the relationships that hold between them in some domain. The aim of ontology is to develop knowledge representations that can be shared and reused.

Ontology is a body of knowledge describing some domain, typically common sense knowledge domain.[7]

2.4 History of Ontology Languages

At the beginning of the 1990s, a set of AI-based ontology implementation languages were created. Following Figure 2. Describes the hierarchy of different ontology languages.[8]

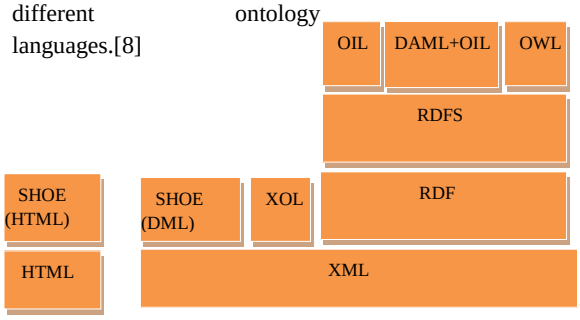

Fig 2: Stack of Ontology Markup Languages

SHOE was built in 1996 as an extension of HTML, in the University of Maryland. It uses set of tags which are different form the HTML specification thus it allows insertion of ontologies in HTML documents. SHOE just allows representing concepts, their taxonomies, n-array relations, instances and deduction rules.

Then XML was created and widely used as a standard language for exchanging information on the web. Then SHOE syntax was modified to includes XML, and some other ontology languages are also built on XML.

XOL was developed by the AI center of SRI international, in 1999. It is a very restricted language where only concepts, concept taxonomies and binary relations can be specified. No inference mechanisms are attached to it. It is mainly designed for the exchange of ontologies in the biomedical domain.

Then RDF was developed by the W3C (The world wide web consortium) as a semantic network based language to describe Web resources. RDFSchema was built by the W3C as an extension to RDF with frame-based primitives. The combination of both RDF and RDFSchema is normally known as RDF(S). RDF(S) is not very expressive. It just allows the concepts, concept taxonomies and binary relations.

Three more languages have been developed as extensions to RDF(S): OIL, DAML + OIL and OWL. OIL was developed in the framework of the European IST project On-To-Knowledge. It adds frame-based Knowledge Representation primitives to RDF(S), and its formal semantics is based on description logics.

DAML + OIL was created by a joint committee from the US and the EU in the con-text of the DARPA project DAML. DAML + OIL also adds DL-based KR primitives to RDF(S). Both OIL and DAML + OIL allow representing concepts, taxonomies, bi-nary relations, functions and instances. Many e orts are being put to provide reasoning mechanisms for DAML + OIL.

Finally, in 2001, the W3C formed a working group called Web-Ontology (WebOnt) Working Group. The aim of this group was to make a new ontology markup language for the Semantic Web, called OWL (Web Ontology Language).

2.5 Domain Ontology

Is an Ontology model which frames definitions and relationships of the concepts, principles, major theories and activities in the domain. It also provides particular meaning of term as they apply to concern domain. For instance the word mouse has different meaning when used in terms of computer and otherwise. And it is of utmost importance to appreciate this difference. Domain Ontology facilitates shared and common understanding of a specific domain.

2.6 Mining based Automatic Ontology Construction

Mining based techniques implement some mining techniques to retrieve the keywords from the given text documents. Mining techniques incorporate automatic key word extraction techniques in order to construct the ontology. Here the text documents can be web pages or files.

2.7 Information Retrieval

In this section we have describe how to retrieved the information using Boolean [9], phrase and category based query.

AND Search: In this type of search, keywords are used to retrieve selected documents for each of the keywords. For example: two key words are used K1 and K2, then their document set D = {d1,d2,d3,...,dn} and D1= {d11,d12,.,d1m}.

Then the weight calculations of two document sets are made using term frequency and inverse document frequency (tf-idf) .Then the two tf-idfs are added to get the intersection of both documents to get new document set as: D1∩D2= {Dt} = {a1,a2,....an}. every document of this sets of documents contain each keyword in each documents.

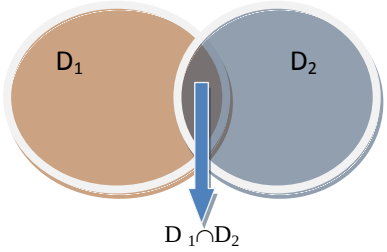

Fig3: AND Search

OR Search: In the or search it is similar to and search, but instead of intersection of D1 and D2, we use union of D1 and D2. It means D1∪D2 is the set of searched document that contain keyword K1 or K2 and both.

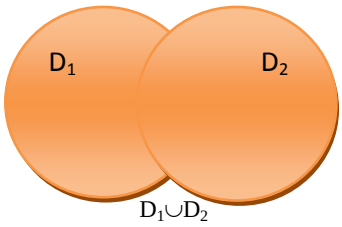

Fig4: OR Search

Phrase Search: In the phrase search , the keywords K1 and K2 are used and their positions are found. Then if Position(k1) +1=Position(k2) then we select the document for further process.

Category Search: In the category search we do indexing for the document body. Index a particular domain in particular category. So it will help the user to select the category for their keyword for fast and appropriate searching. afterwards we can also use AND search or OR search . for example: if user type keyword "thread" and select the category textile then it will fetch the results related to textile. And if user select category computer science then it will fetch the results related sub processes (operating system).

3. ALGORITHM

This paper section describes knowledge about Algorithms and Data-Structure.

3.1 TF-IDF[10]

The formal procedure for implementing a given a documents is collection D, and word w, and an individual document d ∈ D

$$wd = fw,d * \log(|D|/fw,D)$$

where

- fw,d equals the number of times w appears in d,
- |D| is the size of the corpus (number of documents), and
- fw,D equals the number of documents in which w appears in D.

Logarithm of document frequency in the above formula is used for smoothing purpose. The tf-idf value is

- high when a term occurs many times within a small number of documents,
- low when the term occurs fewer times in a document, or occurs in many documents,
- lower when the term occurs in virtually all documents.

Example: Consider a document containing 100 words where in the word \java\ appears3 times. Following the previously defined formulas, the term frequency (TF) for java is then (3 / 100) = 0.03. Now, assume we have 10 million documents and java appears in one thousand of these. Then, the inverse document frequency is calculated as log(10000000 / 1000) = 4. The tfidf score is the product of these quantities: 0.03 4 = 0.12

3.2 TRApriori Algorthim

Inputs: I is set of itemsets , D is multiset of sub set of I
Output: all frequent itemsets and all valid association rules in D

1. Level = 1; frequent_sets = Φ;
2. Candidate_sets = {{i} | i ε I };
3. While Candidate_sets ≠ Φ
4. Scan databade D to compute the frequencies of all sets in candidate_sets

//An itemset A is closed in a data set D if there exists no proper super-itemset B such that B has the same support count as A in D. An itemset A is a closed frequent itemset in set D if A is both closed and frequent in D.

// An itemset A is a maximal frequent itemset (or max-itemset) in set D if A is frequent, and there exists no super itemset B such that A $\subset$ B and B is frequent in D.

5. frequent_sets = frequent_sets $\cup$ { C ε Candidate_sets $\mid$ frequency(C) $\geq$ min_fr};
6. level = level + 1;
7. Candidate_sets = (frequent_sets ε Candidate_sets) $\wedge$ (frequency(C) $\geq$ min_fr and) $\wedge$ ($\mid$ Candidate_sets $\mid$ > level) $\wedge$ (number of combination of Candidate_sets > 1) ;
8. Candidate_sets = { A $\subset$ I $\mid$ $\mid$A$\mid$ = level ans B ε frequent_sets for all B $\subset$A, $\mid$B$\mid$ = level-1};
9. Output frequent_sets;

4. IMPLEMENTATION

This paper section describes our work in the semantic searching engine application. This has been implemented a Semantic Search Engine with re-ranking, Boolean search and category.
The implementation of our system has the following phases:

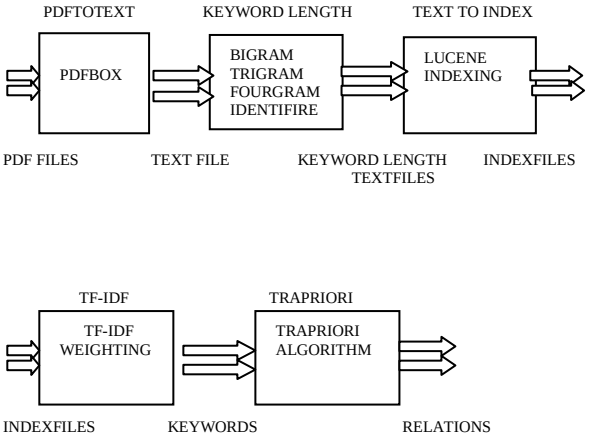

Fig 5. system overview

4.1 Parsing:

Parsing is the method in which we scan whole document for extracting keywords from parse document. we used to parse the Pdf, Ppt, Doc and xls files by pdftotext[11], catppt[12], catdoc[13], xlstocsv[14] to convert them into text (as required by our indexing utility, Lucene[15]). If the pdf file is big like any book then we also use the PDFBox that break the big file in small files.

4.2 Tokenization:

This step extracts word tokens (index terms) from running text. For example, given a piece of text: Java and cpp are very good. it outputs [java, and, cpp, are, very, good]. In tokenization we do not concern comma and full stop symbols

4.3 Stop-word eliminator:

In this step, stop words are removed from the list of tokens. For example, given the list of token generated by tokenizer, it strips it down to: [java, cpp, very, good]. Output show that [and, are] word have removed from tokens list. These word are very common to appear in whole document. Words are removed are store in different table.

4.4 Stemming:

In this step we deal the root or main part of a word to which inflection or formative element are added. For example: A stemming algorithm reduces the words "fishing", "fished", "fish", and "fisher" to the root word, "fish". On the other hand, "argue", "argued", "argues", "arguing", and "argus" reduce to the stem "argu". There are many types of stemming algorithms which differ in respect to performance and accuracy and how certain stemming obstacles are overcome. We use Suffix stripping algorithms in our system for stemming.

4.5 Indexing:

We use Lucene in our system to index the keywords Lucene can index any data that can be converted to textual format, and make it searchable. And it adds searching capability in application. In this paper we used to keyword-by-documents table to represent indexed documents where each row contains the number of occurrences of the appropriate keyword in the appropriate document.

4.6 Inverted indexing:

The ordinary index would contain for each document, the index terms within it. But the inverted index stores for each term the list of documents where they appear. The benefit of using an inverted index comes from the fact that in information retrieval we are interested in finding the documents that contain the index terms in the query. So, if we have an inverted index, we do not have to scan through all the documents in collection in search of the term. Often a hash-table is associated with the inverted index so that searching happens in O(1) time. Inverted index may contain additional information like how many times the term appears. in the document, the o set of the term within the document etc.

Example: Example Say there are three documents

Doc1- cpp is good.
Doc2- Java and cpp are very good.
Doc3- Php is better.

After stop-word elimination and stemming, the inverted index looks like-

good 1,2
cpp 1,2
java 2
php 3

4.7 Querying:

It is given the keyword string, it is the process of searching the best matching previously indexed documents. This process is follow into two part

First: Match the document to the keyword string.
Second: Then result display in relevant order matched the documents (Re-ranking)

4.8 Matching:

Matching is the process of create the list of documents therefore match the query. This is done by string

matching between textual content and ontological data. Here two assumptions are made (1) word synonyms are considered through the use of WordNet synonym sets. (2) Multiword terms will undergo word-level matches. For example capital-city is considered as the synonym of both capital and city.

4.9 Re-ranking:

In this paper we have used to modified TF-IDF algorithm for re-ranking. therefore given the match document "d", weight of keyword \t\ in this document with respect to frequency of "t" in the current document and also get the frequency of t in all documents. Note that if the keyword occurred only one document therefore keyword weight is get zero weigh.

5. EVOLUTION

In this section we compared the results obtained by two systems. In this paper we have done configuration in Joomla Custom Google Search Engine API [16] and our existing search engine for the Evaluation purpose. Figure 6 and 7. We have taken URL (www.it.iitb.ac.in/Cportal) and we have follow all steps like parsing, indexing etc. So we have got the some results based on No. of result, Time same query type by the user. The search keyword (3) "project" by the user. Some experiment result is show the below. In this case we have seen the result of custom google search, it display only 2000 results (0.35 sec) but it is not showing the most relevant keyword (project). Our system display figure.7 the related keyword "project" with 6060 results (0.22 sec) based on ranking. In table we present the number of results and time for some keywords that evaluate by custom google search engine and our portal search engine.

Table1: show results in time for custom google search and our portal system.

KEYWORDS	CUSTOM GOOGLE SEARCH (G)	OUR PORTAL SEARCH (P)
ROM (1)	2400 RESULTS 0.10 SECOND	5800 RESULTS 0.09 SECOND
NETWORKING (2)	280 RESULTS 0.06 SECOND	1020 RESULTS 0.05 SECOND
PROJECT (3)	2000 RESULTS 0.35 SECOND	6060 RESULTS 0.22 SECOND
JAVA (4)	980 RESULTS 0.16 SECOND	5040 RESULTS 0.12 SECOND

Custom Google Search Results and Our system search Results

Fig 6: Custom Google Search Results

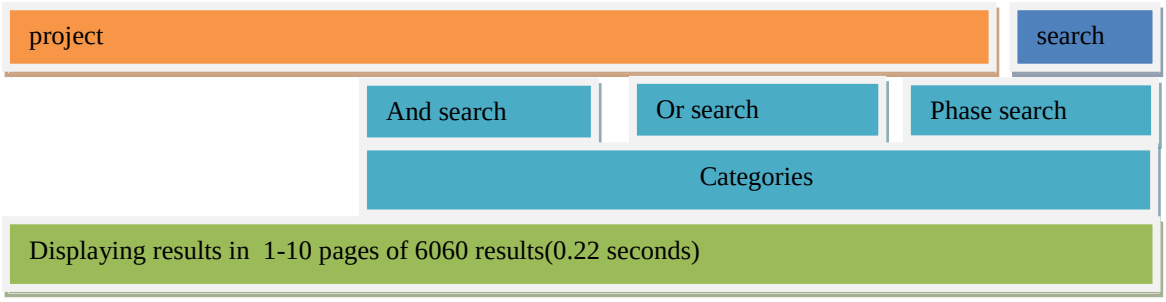

Fig 7: our system Search Results

The results are also show in bar graph for normal search, for AND search and for OR search.

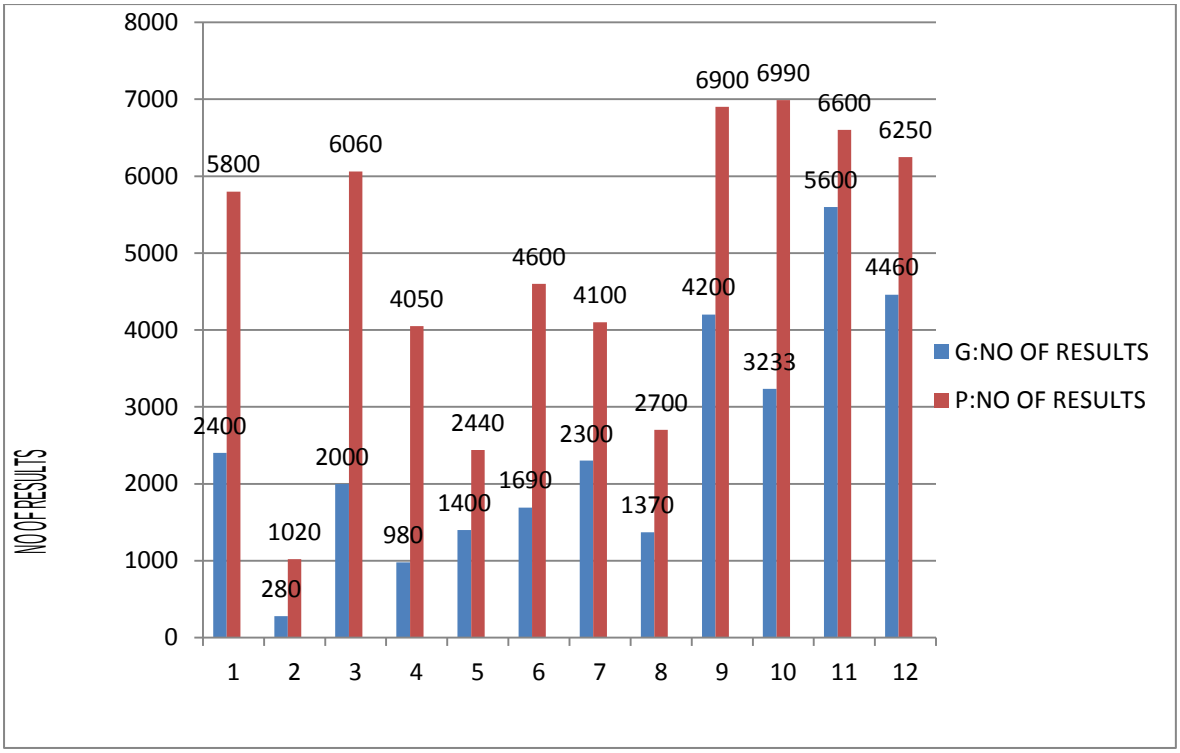

Figure 8: Evolution

Table2: show results in time for custom google search and our portal system . for AND Search

KEYWORDS	CUSTOM GOOGLE SEARCH (G)	OUR PORTAL SEARCH (P)
ADVANCE JAVA (5)	1400 RESULTS 0.14 SECOND	2440 RESULTS 0.11 SECOND
COMPLET PROCESSES (6)	1690 RESULTS 0.23 SECOND	4600 RESULTS 0.12 SECOND
FINITE AUTOMATA (7)	2300 RESULTS 0.09 SECOND	4100 RESULTS 0.06 SECOND
USER FRIENDLY (8)	1370 RESULTS 0.19 SECOND	2700 RESULTS 0.14 SECOND

Table3: show results in time for custom google search and our portal system . for OR Search

KEYWORDS	CUSTOM GOOGLE SEARCH (G)	OUR PORTAL SEARCH (P)
ADVANCE JAVA (9)	4200 RESULTS 0.19 SECOND	6900 RESULTS 0.14 SECOND
COMPLET PROCESSES (10)	3233 RESULTS 0.25 SECOND	6990 RESULTS 0.30 SECOND
FINITE AUTOMATA (11)	5600 RESULTS 0.35 SECOND	6600 RESULTS 0.24 SECOND
USER FRIENDLY (12)	4460 RESULTS 0.22 SECOND	6250 RESULTS 0.18 SECOND

6. PERFORMANCE

G : no of results by Custom Google Search Engine
P : no of results by our portal (our system)
Now we calculate the performance of these search engine on the basis of number of results per second. We take 12 keywords 1 to 4 for normal search and 5 to 8 for AND search and 9 to 12 for OR search then we find the following results . our system is faster 2.6859 times then custom google search engine for the keyword ROM in the particular domain. For each keyword from table 1,2,3 the performance of our portal search is better than custom google search engine. All results are show in table 4.

Table4: performance

KEYWORDS	CUSTOM GOOGLE SEARCH (G)	OUR PORTAL SEARCH (P)
ROM (1)	1	2.68513
NETWORKING (2)	1	4.37145
PROJECTS (3)	1	4.82069
JAVA (4)	1	6.85712
ADVANCE JAVA (5)	1	2.21817
COMPLET PROCESSES (6)	1	5.21698
FINITE AUTOMATA (7)	1	2.67367
USER FRIENDLY (8)	1	2.67459
ADVANCE JAVA (9)	1	2.22952
COMPLET PROCESSES (10)	1	1.80173
FINITE AUTOMATA (11)	1	1.71875
USER FRIENDLY (12)	1	1.71275

7. CONCLUSION

In this portal we have implemented a semantic search engine. This semantic search engine can search all given hyperlink with HTML pages and documents such as PDF, PPT, DOC, CSV les. We use TRapriori algorithm while indexing. We maintain keyword ontology generated with TRapriori algorithm. We have done indexing and re-indexing all repository and also re ne the duplicate data.

8. FUTURE WORK

Since purpose of the C-Portal is to search educational videos, we can go one step ahead to suggest user more videos that will also be useful for him/her which will act as complementary material. We can implement this feature using previous users' search history. User should be able to filtering results as per resource media category like videos, PDFs, PPTs. Its also possible to analyze difficult topic/sub-topic of many users based on responses entered by previous users. This can help educational content maker or speaker to emphasis on particular topic. Also we can recognize upcoming trends among students. Here the type of documents in the database are pdfs, ppts and text document, but videos are not converted to texts. In future the videos can also be stored in the database along with pdfs and ppt. Then more documents will be available. The text is extracted from the videos by voice to text conversion and then when text is extracted, it is same as other documents.

9. REFERENCES

[1]. WebReferenchttp://ekalavya.it.iitb.ac.in/EectiveTeaching Course.do DownloadDate : 21-jun-2011

[2]. WebReferenchttp://www.joomla.org/download.html, here we download joolma framework exe.

[3]. WebReferenchttp://download.cnet.com/XAMPP/3000-10248_4-10703782.html

[4]. MIT,"Mitopencourseware-monthlyreports," accessed 16-February-2011.[Online].Available: http://ocw.mit.edu/about/site-statistics/monthly-reports/

[5]. NPTEL, "Nptel— project document," Department of Secondary and Higher Education, Ministry of Human ResourceDevelopment,GovernmentofIndia,NewDelhi., July 2007.

[6]. WebReferencehttp://en.wikipedia.org/wiki/Model view controller Accessed on 10 March 2011

[7]. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud., 43:907928, December 1995

[8]. Oscar Corcho, Mariano Fernndez-lopez, and AsunciOnGOmez-perez. Methodologies, tools and languages for building ontologies: Where is their meeting point. Data and Knowledge Engineering, 46:4164, 2003

[9]. George Boole. An Investigation of the Laws of Thought on which are founded the Mathematical Theories of Logic and Probabilities. Macmillan and Co, London, 1854.

[10]. Juan Ramos. Using tf-idf to determine word relevance in document queries. First International Conference on. Machine Learning, 2003.

[11]. WebReferenchttp://www.download3k.com/Install-Ease-Pdf-to-Text-Extractor.html

[12]. Web Referenchttp://ftp.findthatfile.com/search-2897383-fEXE/software-tools-download-catppt.exe.htm

[13]. Web Referenchttp://ftp.findthatfile.com/search-774018-fEXE/software-tools-download-catdoc.exe.htm

[14]. WebReferenchttp://www.download25.com/install/batch-xls-to-csv-converter.html

[15]. Wikipedia, "Lucene— wikipedia, the free encyclopedia," 2011, [Online; accessed 24-march-2012].[Online].Available: http://apache.techartifact.com/mirror/lucene/java/3.6.0

[16]. Web Referencehttp://www.google.com/cse/ , for searching keywords by custom google search engine.