



UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING & INFORMATICS
MSc. IS PROJECT

**AN INTELLIGENT SUBTOPICS SUGGESTER FOR AN E-LEARNING
SYSTEM**

By

GLADYS W KING'ORI

SUPERVISOR

Dr. ENG. LAWRENCE MUCHEMI

A research submitted to the School of Computing and Informatics in partial fulfillment of the requirements for the award of Masters of Science in Information Systems of the University of Nairobi.

AUGUST 2014

Declaration

This project is my original work and to the best of my knowledge, it has not been presented for a degree in any other university.

Gladys Wangechi Kingøori

P56/70746/2007

Signed

Date

This project has been submitted as part fulfillment of requirements for the award of Masters of Science in Information Systems of the School of Computing and Informatics of the University of Nairobi, with my approval as the University Supervisor.

Dr. Eng. Lawrence Muchemi

Signed

Date

Abstract

eLearning, is a planned teaching or learning experience that uses a wide spectrum of computer-based technologies to reach learners. The focus of this project was the case where students are logging into and using the system independently of other students and staff members. This fits firmly into the general definition of the asynchronous e-learning environment, where the instructor develops content and the students access it online. The instructor also suggests content to students with specific needs, to do this, current systems use face-to-face discussions, online discussions or emails. Because of the increased number of students involved in eLearning this individualized attention to students' needs has become untenable and unsustainable hence the need for an Intelligent Subtopics Suggester. The model lacks, from a literature point of view. This research aimed at addressing this gap through design of an Intelligent Subtopics Suggester model.

This project developed an Intelligent Subtopics Suggester model that is appended to an eLearning system and that analyzes the user's questions and suggests help subtopics to the lecturer/tutor. The intelligent model was based on a knowledge base that was made using knowledge from domain experts. The knowledge base captured the keywords and terminologies that describe a subtopic of interest. These keywords and terminologies were matched against the keywords in the students' question. This helped to identify the students' key learning needs, by identifying the subtopics learners have not understood well, based on the frequency that a question is asked from a given subtopic.

Based on the above model a prototype was developed, the prototype has 6 production rules for inference and 1505 facts (keywords) in its knowledge base.

The Intelligent Subtopic Suggester was able to identify the topics from which questions had been asked. To evaluate the prototype three accuracy standards were applied, these were precision, recall and accuracy, on average the Intelligent Subtopic Suggester has a precision of 0.727, a recall of 0.972 and an accuracy of 0.766. The Intelligent Subtopic Suggester was found to have acceptable values of precision and accuracy and hence can be considered reliable.

The findings of this research will be of great importance to eLearning system developers and the research community.

Keywords: Intelligent Subtopic Suggester, eLearning, Intelligent tutor systems, Artificial intelligence, web based learning, precision, recall, accuracy.

Acknowledgements

First and foremost, I thank our Almighty God, who makes everything possible. This project would not have been possible without the help of many others. I acknowledge the enormous input by my project supervisor, Dr. Eng. Lawrence Muchemi whose guidance, corrections and suggestions throughout this project were invaluable.

I wish to acknowledge my children, Ann Tracy and Oscar Patrick for their understanding and support during the entire study. I acknowledge my Parents for being great role models and teaching me to never give up.

To my colleagues thank you for your encouragement and support in various stages of the project. I wish to appreciate others whom I have not mentioned here, who contributed to this project in one-way or another. May God bless you all.

Dedication

I dedicate this work to my Parents Joseph Kingøri and Elizabeth Wanjiru

Table of Contents

DECLARATION	I
ABSTRACT	II
ACKNOWLEDGEMENTS	III
DEDICATION	IV
LIST OF FIGURES	VIII
LIST OF TABLES	IX
LIST OF GRAPHS	X
ABBREVIATIONS AND ACRONYMS	XI
CHAPTER ONE INTRODUCTION	1
1.1 BACKGROUND TO THE STUDY	1
1.1.1 <i>Information retrieval</i>	1
1.1.2 <i>E-Learning</i>	1
1.1.3 <i>Intelligent Tutoring System (ITS)</i>	4
1.2 PROBLEM DEFINATION	5
1.3 OBJECTIVES OF THE STUDY	6
1.4 SIGNIFICANCE OF THE STUDY.....	6
1.5 SCOPE.....	7
CHAPTER TWO LITERATURE REVIEW	8
2.0 INTRODUCTION	8
2.1 STRING SEARCHING ALGORITHMS.....	9
2.1.1 <i>Naïve string search (Brute-force algorithm)</i>	9
2.1.2 <i>Boyer-Moore-based approach</i>	10
2.1.3 <i>The Knuth-Morris-Pratt Algorithm</i>	12
2.1.4 <i>Aho-Corasick algorithm</i>	13
2.1.5 <i>Rabin-Karp string search algorithm</i>	14
2.1.6 <i>Suffix Trees</i>	14
2.2 PATTERN MATCHING.....	15
2.2.1 <i>Pattern Matching using Regular Expressions</i>	15
2.2.2 <i>PostgreSQL LIKE operator</i>	16
2.2.3 <i>Full text search</i>	16
2.3 INTELLIGENT TUTORING SYSTEMS MODELS.....	18
2.3.1 <i>Domain Knowledge Module</i>	18

2.3.2 Teacher Module.....	18
2.3.3 Student Module.....	19
2.3.4 Interface Module.....	19
2.3.5 Student Module.....	19
2.3.6 Domain Module	19
2.3.7 Expert Module.....	20
2.3.8 Pedagogical module.....	20
2.3.9 Communication module	20
2.4 THE CONCEPTUAL MODEL.....	20
2.4.1 User Interface Module.....	21
2.4.2 Intelligent Subtopic Suggester	21
CHAPTER THREE METHODOLOGY	22
3.0 INTRODUCTION	22
3.1 RESEARCH METHODOLOGY	22
3.2 ISSUES THAT NEED TO BE ADDRESSED	22
3.4 THE INTELLIGENT SUBTOPIC SUGGESTER PROCESS FLOW	25
3.5 DETAILED DESIGN OF THE COMPONENTS.....	26
3.5.1 Pattern Matcher Design	26
3.5.2 Question Analyzer (QA) design	27
3.5.3 Subtopic ranker Design	30
3.5.4 Knowledge Base Design (db2).....	33
3.6 PROTOTYPE DEVELOPMENT APPROACH.....	33
3.6.1 SELECTION OF THE PROTOTYPE'S DEVELOPMENT TOOLS.....	34
3.6.3 The User Interface Design	36
3.2.3.3 THE LECTURERS INTERFACE	37
3.7 IMPLEMENTATION OF THE MODULES.....	38
CHAPTER FOUR PROTOTYPE EVALUATION.....	40
4.0 INTRODUCTION	40
4.1 DATA COLLECTION FOR THE TEST.....	42
4.2 CLEANING THE DATA.....	42
4.3 DATA SAMPLING.....	42
4.4 MODE OF ANALYSIS.....	43
4.5 RESULTS.....	43

4.5.1 Summary of the results of the data from Internet.....	43
4.5.2 Summary of the Results of the Data from Students and DE.....	44
4.6 RESULTS ANALYSIS.....	44
4.7 DISCUSSION OF RESULTS.....	46
CHAPTER FIVE CONCLUSION AND RECOMMENDATIONS.....	48
5.0 CONCLUSION.....	48
5.2 LIMITATIONS OF THE STUDY	49
5.3 RECOMMENDATIONS.....	49
5.4 SUGGESTIONS FOR FURTHER RESEARCH.....	50
REFERENCES	51
APPENDIX A: SAMPLE CODE	54
APPENDIX B: INTERVIEW CHECK LIST	69
APPENDIX C: USER MANUAL.....	70
APPENDIX D: STOPWORDS	71

LIST OF FIGURES

- Figure 2.1 Architecture of ITS (Stankov, Glavinic & Rosic, 2000). **Error! Bookmark not defined.**
- Figure 2.2 Architecture of its (Woolf, 1992)..... **Error! Bookmark not defined.**
- Figure 2.3 Conceptual model for the Intelligent Subtopic suggester **Error! Bookmark not defined.**
- Figure 3.0 Conceptual model for the Intelligent Subtopic suggester **Error! Bookmark not defined.**
- Figure 3.1 Components of the Intelligent Subtopic suggester..... **Error! Bookmark not defined.**
- Figure 3.2 Architectural Design of the Intelligent Subtopic suggester **Error! Bookmark not defined.**
- Figure 3.3 The Process Flow for the Intelligent Subtopic Suggester **Error! Bookmark not defined.**
- Figure 3.4 screen shot of the data in the QA after pattern matching **Error! Bookmark not defined.**
- Figure 3.5 A screen shot showing the relevance of the returned result. **Error! Bookmark not defined.**
- Figure 3.6 The Prototyping Process..... **Error! Bookmark not defined.**
- Figure 3.7 The student interface **Error! Bookmark not defined.**
- Figure 3.8 A screen shot of the domain experts' interface. **Error! Bookmark not defined.**
- Figure 3.9 A screen shot of the lecturers' interface. **Error! Bookmark not defined.**

LIST OF TABLES

Table 3.1 student module Unit testing**Error! Bookmark not defined.**

Table 3.2 DE module testing**Error! Bookmark not defined.**

Table 3.3 Courseoutline table**Error! Bookmark not defined.**

Table 3.4 Analyzer Table**Error! Bookmark not defined.**

Table 3.5 Student Record table.....**Error! Bookmark not defined.**

Table 4.1 Results comparing performance of the QA when used with questions from internet
.....**Error! Bookmark not defined.**

Table 4.2 Performance of the questions from the students and domain experts table.....**Error!
Bookmark not defined.**

Table 4.3 Performance standards of the Intelligent Subtopic Suggester on Internet data**Error!
Bookmark not defined.**

Table 4.4 Performance standards of the Intelligent Subtopic Suggester on students and domain
experts data.....**Error! Bookmark not defined.**

LIST OF GRAPHS

Graph 4.1 Performance standards of the Intelligent Subtopic Suggester on Internet data.....**Error! Bookmark not defined.**

Graph 4.2 Performance standards of the Intelligent Subtopic Suggester on students and domain experts data.....**Error! Bookmark not defined.**

ABBREVIATIONS AND ACRONYMS

Artificial Intelligence (AI) this is the study and creation of computer systems that can perceive, reason and act

Intelligent System (IS) this is a system that emulates some aspects of intelligence exhibited by nature. These include learning, adaptability, robustness across problem domains, improving efficiency (over time and/or space), information compression (data to knowledge), extrapolated reasoning.

Expert System (ES) this is Expert system is program that has Artificial Intelligence expert-level Knowledge about a particular domain and knows how to use its Knowledge to respond properly.

A Knowledge Based Systems (KBS) this is a system that has the human expert knowledge in a formalized and structured form.

Intelligent Tutor System (ITS) this is an educational software containing an artificial intelligence component. It is any computer system that provides direct, customized instruction or feedback to students.

Domain expert (DE) this is a knowledgeable and skilled person who has the greatest expertise in a given domain

CHAPTER ONE INTRODUCTION

1.1 BACKGROUND TO THE STUDY

1.1.1 Information retrieval

Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. Automated information retrieval systems are used to reduce "information overload". Many universities and public libraries use Information retrieval systems to provide access to books, journals and other documents. Web search engines are the most visible Information retrieval applications.

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy.

A typical information retrieval task is to select documents from a database in response to a user's query, and rank these documents according to relevance, (Strzalkowski & Vauthey, 1992).

1.1.2 E-Learning

E-Learning is one of the tools used in the knowledge management to share knowledge among groups (Veeramani, 2010). The term eLearning is used in literature and commercial applications to describe many fields such as online learning, web-based training, distance learning, distributed learning, virtual learning, or technology-based training (Veeramani, 2010), (Alkhatabi, Neagu, & Cullen, 2010). It is the online delivery of information, communication, education, and training providing a new set of tools that can add value to all the traditional learning modes, such as classroom experiences, textbook study, CD-ROM, and traditional computer based training, (Omwenga & Rodrigues, 2006).

It involves the use of personal computers, CD-ROMs, Digital Television, and Mobile Phones. Communications technology enables the use of the Internet, email, discussion forums, collaborative software and team learning systems, (Veeramani, 2010).

Electronic learning, sometimes called distance learning, therefore is a planned teaching or learning experience that uses a wide spectrum of technologies, mainly Internet or computer-based technologies to reach learners (Veeramani, 2010). The definition of e-learning centers on it being a learning method and a technique for the presentation of academic curricula via the

Internet or any other electronic media inclusive of multimedia, compact discs, or other new education technologies. The two parties participating in the educational process interact through these media to achieve specific educational objectives.

There are two main types of eLearning: Synchronous e-learning environments require tutors and learners or the online classmates to be online at the same time, where live interactions take place between them (Alkhatabi, Neagu, & Cullen, 2010). Software tools for synchronous eLearning include audio conferencing, video conferencing, and virtual whiteboards that enable both instructors and students to share knowledge. In an asynchronous learning environment, the instructor only interacts with the student intermittently and not in real time.

Asynchronous learning environment is one in which an instructor teaches a somewhat traditional class but the instructor and students are online simultaneously and communicate directly with each other. Asynchronous learning is supported by such technologies as online discussion groups, email, and online courses (Veeramani, 2010). Asynchronous Learning Network is a variety of e-learning systems, which distribute learning materials and concepts in one direction at a time. The learners interact with learning materials, tutors and other learner/s through the Internet at different times and from different places, (Strzalkowski & Vauthey, 1992).

In the new economy, organizations and academic institutions need to find new, cost-effective ways to keep the workforce current and competent. Knowledge and skills have to be delivered more rapidly and efficiently whenever and wherever needed to account for the increasing complexity and velocity of the work environment. In the age of just-in time production, just-in-time training has become a critical element of organizational success, (Veeramani, 2010).

Social and demographic changes, such as declining birth rates, the aging population, and shortage of skilled labor, are directing education toward new target groups. Today, traditional students, age 18 to 22, represent only a fraction of the higher education student population. Working adults, the fastest-growing group attending higher education institutions, are excellent candidates for education delivered to their homes or offices. The explosive growth of the Internet opens up opportunities to support demographic, technological, and lifestyle changes and offer quality education to those who would otherwise not have access to it.

A primary task of the teacher is to provide feedback. In E-learning, teacher's feedback or comments are in email messages. They usually correct the original file submitted by the student and students can download it to see the comments, (Veeramani, 2010).

Most universities have implemented e-learning technologies and this has been done by incorporation of open source and commercial based e-learning platforms. However, despite the implementation of these platforms as content delivery channels, students have different perceptions in regard to suitability of the platform and interactivity of these very platforms, (Odhiambo & Omondi, 2009).

As examples, a case study of Jomo Kenyatta University of Agriculture and Technology and United States International University which uses moodle e-learning platform and WebCT e-learning platforms respectively, revealed that students from both Jomo Kenyatta University of Agriculture and Technology and United States International University have high perceptions that e-learning platforms implemented in their institutions were not interactive and have low usability capabilities, (Odhiambo & Omondi, 2009).

According to the Kemu website, they offer a Distance Learning Mode that is a self paced mode of study in which the student is provided course materials for study, and may have additional material available to them online via the KeMU DLM Portal.

The University of Nairobi, has implemented three different learning management systems: Wedusoft7, Chisimba8 and Claroline9, (Ssekakubo, Suleman, & Marsden, 2011).

Through the Intel® Explore and Learn online platform, learners from Kenya, South Africa and Nigeria can find and acquire free and paid for, textbooks, instructional videos, exam preparation materials to help them in their studies, with content ranging from officially approved text books and multimedia content to supplementary multimedia and interactive content from local and international providers all designed to enhance the learning experience for students from three African countries, Kenya, South Africa and Nigeria, (Ouma, 2013).

The aim of the Intel® Explore and Learn solution is to provide a resource hub where learners can find relevant content to enrich their studies while at the same time provide publishers with access to a platform that will enable them to easily share their learning content. The platform provides access to wide variety of content including text books, set books, revision books, interactive past exam papers, multimedia (audio, video and animations), (Ouma, 2013).

In a study involving a group of undergraduate students at the University of Nairobi doing a bachelor's degree in Computer Science, the students used an eLearning platform called Wedusoft. Evaluation of Wedusoft electronic learning environment using the Technology Mediated Learning Evaluation Framework showed that although the undergraduate students

were generally dissatisfied with online learning, they found quite a few features that they would prefer to have changed or implemented differently, for instance, most of them felt that the material should be presented mainly in point form while others thought that it might be desirable to have the instructor available in near synchronous mode to deal with their queries during the process of learning, which is a demand that would be difficult to implement and could negate the spirit of flexible, individualized learning, (Omwenga & Rodrigues, 2006).

One aspect of e-learning process that often generates considerable concern for teachers is evaluation of student performance. They worry that they will not be able to assess student understanding or participation properly, (Veeramani, 2010).

1.1.3 Intelligent Tutoring System (ITS)

Broadly defined, an intelligent tutoring system (ITS) is any computer system that provides direct, customized instruction or feedback to students. Intelligent tutoring systems can also be viewed as computer software systems that seek to mimic the methods and dialog of natural human tutors, to generate instructional interactions in real time and on demand as required by individual students. Another definition of ITS is given by Hafner who broadly defines, an ITS as an educational software containing an artificial intelligence component, (Hafner, 2004).

There are several categories of ITSs. Some systems based on abstraction of the learning environment provide instruction by simulating a realistic working environment in which the student can learn the task. An example of a simulation-based ITS is the Advanced Cardiac Life Support (ACLS) Tutor (Eliot & Woolf, 1995) in which a student takes the role of a team leader in providing emergency life support for patients who have had heart attacks. The system monitors the student's actions, and also runs a realistic simulation of the patient's condition and maintains an environment that is reasonably faithful to the 'real life' situation. The goal is to test the student's knowledge about the correct emergency procedures, and also to allow him to experience practicing those procedures in a more realistic manner than is possible in a traditional classroom.

Smithtown (Shute, Glaser, & Raghaven, 1989) presents a situation similar to the real world scenarios in which the knowledge could be applied, but they are not exact simulations. Smithtown provides a simulated setting for students to test hypotheses about economics.

However, the underlying model of the environment is not an exact simulation of how the laws of economics would be applied in the real world.

At the extreme opposite of the simulation based tutors are those that teach knowledge in a decontextualized manner without attempting to simulate the real world. These systems provide problems for the learner to solve without trying to connect those problems to a real world situation and are designed to teach abstract knowledge that can be transferred to multiple problem solving situations.

Most ITSs teaches procedural skills with an aim for students to learn how to perform a particular task. An example of a 'cognitive tutor' is SHERLOCK, which has tutorial actions associated with each state in the 'effective problem space' (Lajoie & A, 1992). Another example of an ITS that uses an analysis of expert behavior is the LISP tutor, which encodes expert problem solvers' actions as production rules, and attempts to determine which rules the student is having difficulty applying.

The Scheme Tutor for Analogical ERror EXplanation, (STAEREX) is a prototype tutor system of an Intelligent Tutoring System that assists a learner during solving programming exercises in the functional programming language SCHEME by displaying an example that has been solved correctly in the past, (Pollack & Ute, 2007).

Another type of an ITS is the Intelligent Subtopic Suggester which analyzes the user's questions and suggests help subtopics to the lecturer/tutor. The intelligent model is based on a knowledge base that was made using knowledge from domain experts. The knowledge base captures the keywords and terminologies that describe a subtopic of interest. These keywords and terminologies are matched against the keywords in the students' question. This helps to identify the students' key learning needs, by identifying the subtopics learners have not understood well, based on the frequency that a question is asked from a given subtopic.

1.2 PROBLEM DEFINATION

E Learning, is a planned teaching or learning experience that uses a wide spectrum of technologies, mainly Internet or computer-based technologies to reach learners. The e Learning systems in use today are either Synchronous or Asynchronous e Learning environments. The focus of this project was the case where students are logging into and using the system independently of other students and staff members. This fits firmly into the general definition of

the asynchronous e-learning environment, where the instructor develops content and the students access it online.

With the current eLearning platforms, whenever a student has a problem understanding the content given to them, the students either communicates to the instructor through emails and gets instructor's feedback or comments in email messages, or makes an appointment for a face to face discussion.

Because of the increased number of students involved in eLearning these methods have become untenable and unsustainable hence the need for an Intelligent Subtopics Suggester. From a literature point of view, such a model lacks. This research aimed at addressing this gap through design of an Intelligent Subtopics Suggester model.

1.3 OBJECTIVES OF THE STUDY

The main objective of this project was to identify the students key learning needs, by identifying the subtopics learners have not understood well for given eLearning materials, to suggest to the tutors/lecturers subtopics that students may need further help with and to develop and test a prototype that addresses the above objectives.

The specific objectives were to

- (i) Design an intelligent suggestion model that integrates Knowledge from the necessary knowledge sources and recommends to the tutor/lecturer subtopics that are not well understood based on the frequency of questions from the specific subtopics.
- (ii) Design and develop a prototype based on the above model to demonstrate some of the functionalities.
- (iii) Evaluate the performance of the model through results obtained from the prototype.

1.4 SIGNIFICANCE OF THE STUDY

Web-based education systems have a lot of advantages, however they lack the presence of a teacher, who in a traditional classroom set up, employs various mechanisms to sustain the student's attention, and provides appropriate guidance to the student based on his/her weakness and strengths in a particular subject. However, the material provided for eLearning does not provide for tutorial support to the student.

The Intelligent Subtopic Suggester will be of great significance to such students and their lecturers since it will be possible for them to get improved content for subtopics that they find difficult to understand. The lecturer will also be able to evaluate his eLearning content through the results of the suggester.

This research is also very significant to the eLearning developers who can add the suggester component to their eLearning platforms. The research has also added to the body of knowledge of the research community

1.5 SCOPE

Due to the limitation of time, a model for the Intelligent Subtopic Suggester was designed and a prototype based on it developed. A full system was however not possible.

CHAPTER TWO LITERATURE REVIEW

2.0 INTRODUCTION

According to Feldman, Children learn language by discovering patterns and templates. We learn how to express plural or singular and how to match those forms in verbs and nouns. We learn how to put together a sentence, a question, or a command. Natural Language Processing assumes that if we can define those patterns and describe them to a computer then we can teach a machine something of how we speak and understand each other, (Feldman, 1999).

Natural Language Processing pursues the elusive question of how we understand the meaning of a sentence or a document. What are the clues we use to understand who did what to whom, or when something happened, or what is fact and what is supposition or prediction? While words, nouns, verbs, adjectives and adverbs, are the building blocks of meaning, it is their relationship to each other within the structure of a sentence, within a document, and within the context of what we already know about the world, that conveys the true meaning of a text, (Feldman, 1999)

In this project, natural language processing will be done using pattern matching. Pattern matching can be defined as the comparison of two patterns to determine whether they match, that is, that they are the same or do not match that is that they differ. Testing consists of matching an observed pattern with an expected pattern and deciding whether these patterns match or do not match (resulting in a disconfirmation). In pattern matching the expected pattern is precisely specified before the matching takes place. A pattern is any arrangement of objects or entities, (Singla & Garg, 2012).

Pattern matching is the act of checking a perceived sequence of tokens for the presence of the constituents of some pattern. That is Pattern matching searches a string containing text or binary data for some set of characters based on a specific search pattern. When one searches for a string of characters in a file using the -Findø command in word processor, or when one uses a search engine to look for something on the Web, one is using a simple version of pattern matching: in this case the criteria is "find these characters." In those environments, one can customize the criteria in particular ways, for example, to search for this or that, to search for this or that but not the other thing, to search for whole words only, or to search only for those words that are 12 points and underlined, etc.

String searching is concerned in finding the occurrences of a substring (called a *pattern*) of length **m** in a *text* of length **n**, (Mohammad, Saleh, & Abdeen, 2006). String searching is an

important step towards solving many problems, including text editing, text searching and symbol manipulation. In order to search for a pattern within a string, an algorithm is needed to find the pattern, determination of the algorithm to use depends on the application where the algorithm is to be used (Mohammad, Saleh, & Abdeen, 2006). For example, Word processing applications make use of single string search in its search and-replace operations, (Alqadi, Aqel, & El Emary, 2007).

String searching algorithms are algorithms that try to find a place where one or several strings (i.e. patterns) are found within a larger string or text. Single-pattern string matching, involves a large body of work in which a single string is to be searched for in the text. On the other hand, the multi-pattern string matching problem searches a body of text for a *set* of strings (patterns), preprocesses the set of input strings, and then searches all of them together over the body of text, (Alqadi, Aqel, & El Emary, 2007).

String searching algorithms, sometimes called string matching algorithms, are algorithms that try to find a place where one or several strings (also called patterns) are found within a larger string or text.

The problem of pattern matching is encountered in numerous arenas, from Internet searches to library catalogs. Exact pattern matching involves finding all occurrences of a pattern P in a string S , where S is longer than P .

While exact matching provides a fundamental basis for string searching in sequences, it is important that differences in sequences be accommodated and utilized. This is accomplished through Inexact Pattern Matching, the challenge in inexact or approximate matching, is allowing enough edits (insertions, deletions, and substitutions) to detect relevant patterns, but not so many as to make the comparison irrelevant. With inexact matching, the concern shifts to alignment of sequences based on their similarity. Some of the algorithms used for string pattern matching are:

2.1 STRING SEARCHING ALGORITHMS

2.1.1 Naïve string search (Brute-force algorithm)

It is an exact string matching algorithm, it involves finding all occurrences of a pattern P in a string S , where S is longer than P . According to Wikipedia, this algorithm compares the pattern and text character by character. The first letter of pattern P is lined up with the first letter of strings, and the letters of the aligned region are compared until all of P is found to match the

corresponding S region or a mismatched letter is found, in which case P is shifted one letter to the right and the process is repeated until a match is found or the end of the text is reached. By moving one character at a time, the worst-case time required to compare the pattern to the entire sequence would be proportional to the length of P multiplied by the length of S .

By performing an initial preprocessing of the text, the speed of the main search can be increased appreciably. For example, to find the pattern "MOUSE" in the following string:

MOPMOUSEMOUNTAINMONKEY one could first perform a search locating the 'M' in the string, and then only compare "MOUSE" to those locations.

In a case of no matches found during preprocessing, time is saved because no further searching is needed, and thus the time saved is proportional to the relative sizes of the prefix and the original pattern.

By moving one character at a time, the worst-case time required to compare the pattern to the entire sequence would be proportional to the length of P multiplied by the length of S , that is it has a worst-case complexity of $O(mn)$, where m denotes the length of p and n denotes the length of t , (Khan & Pateriya, 2012), (Tiwari & N, 2012).

2.1.2 Boyer-Moore-based approach

The Boyer-Moore algorithm works by searching the target string from right to left. It is considered one of the most efficient algorithms for general pattern matching applications, (Khan & Pateriya, 2012), (Tiwari & N, 2012). The execution time of the Boyer-Moore algorithm can be sub-linear since the algorithm doesn't need to check every character of the string to be searched, but rather skips over some of them. Generally the algorithm gets faster as the key being searched for becomes longer. Its efficiency derives from the fact that with each unsuccessful attempt to find a match between the search string and the text it's searching, it uses the information gained from that attempt to rule out as many positions of the text as possible where the string cannot match.

In the Boyer-Moore algorithm the pattern is shifted from left to right across the text, but comparison is performed from right to left on the pattern. As soon as a mismatch is detected, the pattern is shifted to the right according to one of two key heuristics: The extended bad character rule and the good suffix rule.

To illustrate the operation of these heuristics, suppose that the pattern, P , is aligned at position k of T , and that a mismatch has been detected between the character at position of the pattern that is, $P[i] \neq T[k + i - 1]$. Then let $c = T[k + i - 1]$, the mismatched character of the text, and $t = P[i + 1 : m]$, the suffix of the pattern which matches the corresponding portion of the text. The extended bad character rule proposes that if there is an occurrence of c in P to the left of i , that the pattern be shifted so that the two occurrences of c are aligned. If no such shift is possible, the pattern is shifted completely past the c in the text. The good suffix rule attempts to align the matched suffix, t , with a previous occurrence of t in the pattern (for example, in the pattern “reduced”, the suffix “ed” occurs twice). If there are no other occurrences of t in the pattern, then the pattern is either shifted so that the prefix of the pattern matches a suffix of t in the text, or, if this is not possible, shifted completely past t , (Bell, Matt, Mukherjee, & Adjero, 2002). The Boyer-Moore algorithm checks both of these heuristics at each stage of the matching process; if both shifts are possible, then the maximum is chosen. In this way, Boyer-Moore achieves so-called ‘sub-linear’ performance for most texts, (Adjero, Mukherjee, & Powell, 1997).

Example; Extended bad character rule

CATDOGMOUSECAT

MOUSE 1. O-E mismatch

CATDOGMOUSECAT

MOUSE 2. O in MOUSE shifted to O in DOG, O-E mismatch

CATDOGMOUSECAT

MOUSE 3. O in MOUSE shifted to O in MOUSE

Example; Good suffix rule.

DOGARKCATMOUSECATBATBAT

CATBATBAT 1. Suffix AT matches, but C-B mismatch

DOGARKCATMOUSECATBATBAT

CATBATBAT 2. Shift pattern P until suffix AT with different left letter is found (i.e. CAT).

Suffix AT matches, but C-B mismatch

DOGARKCATMOUSECATBATBAT

CATBATBAT 3. Shift pattern P until suffix AT with different left letter is found (i.e. BAT).

Suffix ATBAT matches, but C-B mismatch

DOGARKCATMOUSECATBATBAT

CATBATBAT 4. Shift pattern P until suffix AT with different left letter is found (i.e. BAT).

This is a match for CATBATBAT

The combination of the two heuristics of the Boyer-Moore algorithm makes the method simple yet powerful. Immediately, searching from right to left with the Bad Character Shift Rule, the potential to skip over a greater number of nonmatching substrings is realized, (Tiwari & N, 2012), (Lee, 2004) .

The Boyer-Moore algorithm would at worst run in linear time (i.e. proportional to the sum of the lengths of P and S), the algorithm tends to have the best performance in practice, as it often runs in sublinear time. The worst case running time is as bad as that of the naive algorithm, (Tiwari & N, 2012).

2.1.3 The Knuth-Morris-Pratt Algorithm

The Knuth-Morris-Pratt algorithm seeks to improve the length of the pattern shift by utilizing information already gathered from searching a string. To accomplish this goal, the algorithm preprocesses the pattern and creates a finite state machine (or automaton). Typically, the information is portrayed in tabular form with δ KNP Next values assigned to each character in the pattern based on the number of spaces the machine moves the pattern if a mismatch is found.

The algorithm then uses the finite state machine to process the string.

Example: KMP Next Table

G C A G A G A G -

-1 0 0 -1 1 -1 1 -1 1 (KMP Next value)

GCATCGCAGAGAG

GCAGAGAG 1. Mismatch at position 4, shift by 3 (number of matches) minus -1. (KMP Next value of δ G at position 4 of P) = 4

GCATCGCAGAGAG

GCAGAGAG 2. Mismatch at position 1, shift 0 - (-1) = 1

GCATCGCAGAGAG

GCAGAGAG 3. Pattern match, shift 8-1 = 7 to continue

Properly implemented, the Knuth-Morris-Pratt algorithm has the potential to look at each letter of the string only once. However, with the table in the previous example, one can see that some letters will be searched twice, namely those in the middle of mismatches, which have KMP Next values that shift the first letter of the pattern to the mismatched position, (Khan & Pateriya, 2012). The worst case running time of this algorithm is linear, i.e., $O(m+n)$, (Tiwari & N, 2012), (Lee, 2004).

2.1.4 Aho-Corasick algorithm

Aho and Corasick algorithm is a multi-pattern string matching algorithm that adopts a finite state automaton to organize multiple pattern strings in order to get searching of multiple patterns done in one pass. It is suited for applications in which we are looking for a large number of keywords in text strings.

The Aho-Corasick algorithm is a kind of dictionary-matching algorithm that locates elements of a finite set of strings (the "dictionary") within an input text. It matches all patterns "at once", so the complexity of the algorithm is linear in the length of the patterns plus the length of the searched text plus the number of output matches. Note that because all matches are found, there can be a quadratic number of matches if every substring matches (e.g. dictionary = a, aa, aaa, aaaa and input string is aaaa).

Informally, the algorithm constructs a trie with suffix tree-like set of links from each node representing a string (e.g. abc) to the node corresponding to the longest proper suffix (e.g. bc if it exists, else c if that exists, else the root). It also contains links from each node to the longest suffix node that corresponds to a dictionary entry; thus all of the matches may be enumerated by following the resulting linked list. It then uses the trie at runtime, moving along the input and keeping the longest match, using the suffix links to make sure that computation is linear. For every node that is in the dictionary and every link along the dictionary suffix linked list, an output is generated, (Tiwari & N, 2012).

When the pattern dictionary is known in advance (e.g. a computer virus database), the construction of the automaton can be performed once off-line and the compiled automaton stored for later use. In this case, its run time is linear in the length of the input plus the number of matched entries. Its time complexity is $O(n)$, regardless of the number of patterns. However, Aho

and Corasick algorithm suffers an intrinsic deficiency for high speed since it does not excavate any heuristics to avoid unnecessary character comparisons, (Khan & Pateriya, 2012), (Lee, 2004)

2.1.5 Rabin-Karp string search algorithm

The Karp-Rabin Random Fingerprint algorithm is a semi-numerical method which utilizes a hashing function to compare whether a substring of equal length to the pattern is a match. The algorithm uses two concepts, hashing and windowing, to find a pattern in the given text. Hashing is implemented by generating a "key" for the pattern to be searched, and then comparing this "key" with the values generated by moving a "pattern-sized" window over the target text. The hashing function is chosen to generate unique values for all possible combinations of alphabets of all lengths. This uniqueness reduces the problem of pattern search to simple mathematical comparison of hash values, (Singh, 2003), (Lee, 2004).

The hashing function preprocesses the string by moving one letter at a time, assigning a value to substrings the length of the pattern. The function computes the hash value by performing an arithmetic calculation on the pre-assigned values of each letter in the substring.

After the hash function preprocessing, the algorithm performs the same hash function on the pattern, and then performs a left-to-right comparison of substring hash values. A good hash function would only yield the same value if the compared strings were indeed a match. For the Karp-Rabin algorithm, with a pattern of 250 characters and a string of 4000 characters, the probability of producing a false positive is at most $p=0.001$ (Gusfield, 1997). An efficient function would also increase calculation speed by recognizing that with each move only one number is added and one number dropped. Because the Karp-Rabin looks at each character in the string twice (once as it enters the hash function and once as it exits), longer string searches are lengthy, especially during the preprocessing stage. Also, when a hash value match is found, the substring needs to be confirmed, possibly adding appreciable time for long patterns which are repeated frequently in the string.

2.1.6 Suffix Trees

The main premise of the suffix tree is that a string can be broken down into as many suffixes as there are characters, and this information can be stored in tree form. In addition, because the

words are stacked alphabetically, suffixes with multiple locations and common prefixes can be readily located.

Example (Implicit suffix tree):

BANANA

- A -->|--> NA -->|| --> NA|--> NANA- B --> ANANA- N -->|--> A|--> ANA

After the tree is created, one can move between locations of a string quickly, regardless of actual distance in the string this however may lead to slower processing in larger trees if the suffixes of interest are located in different areas of memory. In addition, longer trees require a much greater amount of space than the normal sequence. If every suffix is included in the tree, and there are no duplicate suffixes, for a string of length M , $(M/2)*(M+1)$ characters are required. Suffix trees are also able to provide quick insight into repeated patterns and longest shared substrings. However, the construction of the tree does not benefit from dual processors because each letter must be addressed individually and in order. As such suffix trees of appreciable length need to be stored and maintained to be useful, as building large trees on demand is inefficient.

This algorithm tends to run in linear time in most scenarios of practical interest. The worst case running time is as bad as that of the naive algorithm, i.e., $O(mn)$ (Lee, 2004).

Most applications uses Boyer Moore, BMH or KMP algorithms for their effective and efficient functionality and other applications uses the basics of these algorithms for their functionalities as the KMP algorithm has less time complexity and Boyer Moore and BMH algorithms has preprocessing time complexity less. Other algorithms depends upon the type of input and is efficient for certain or particular application (Lee, 2004).

2.2 PATTERN MATCHING

2.2.1 Pattern Matching using Regular Expressions

Pattern Matching can be done using various forms that include the use of Regular Expressions for example, pattern matching in Perl uses regular expressions, often called just *regexes*. A regular expression specifies a search pattern, using metacharacters (which are, or belong to, operators) and character literals. Regular expressions are a powerful text-processing component of the programming languages Java and PERL.

Regular expressions can for example be used; For input data validation, verifying that input is in the right specific format, for example, that email addresses have the right components, extracting

parts of a file that match a specific criteria, (for example, you could extract the headings from a file to build a table of contents, or extract all the links in and HTML file), counting the number of occurrences of a pattern in a string, searching and replacing.

2.2.2 PostgreSQL LIKE operator

Another method that can be used to do pattern matching is using PostgreSQL LIKE operator. A pattern is constructed by combining a string with wildcard characters and use of the LIKE or NOT LIKE operator to find the matches. PostgreSQL provides two wildcard characters:

- ❖ Percent (%) for matching any sequence of characters.
- ❖ Underscore (_) for matching any single character.

If the pattern does not contain any wildcard character, the LIKE operator acts like the equal (=) operator, (Singh, 2003).

MySQL also provides standard SQL pattern matching as well as a form of pattern matching based on extended regular expressions, which uses the REGEXP and NOT REGEXP operators (or RLIKE and NOT RLIKE, which are synonyms), a REGEXP pattern match succeeds if the pattern matches anywhere in the value being tested. This differs from a LIKE pattern match, which succeeds only if the pattern matches the entire value.

2.2.3 Full text search

Before the advent of the search engine, users had to search manually through dozens or hundreds of articles to find the ones that were right for them. Nowadays, we expect the results to come to the user, not the other way around. The goal of all search engines is to find and organize distributed data found on the Internet. The search engine gets the computer to do the work for the user, (Gyorodi, Gyorodi, Pecherle, Gyorodi, & George, 2010).

Full text search was originally created for use with search engines, for example, WebCrawler was the first search engine to provide full text search. AltaVista, became the first searchable full-text database on the World Wide Web with a simple interface, while Google was the first to use the concept of a search engine based on relevancy ranking.

According to MySQL 5.1 Reference manual, Full Text Searching provides the capability to identify natural-language *documents* that satisfy a *query*. There are three types of full-text

searches:

Natural language full-text search interprets the search string as a phrase in natural human language. By default or with the `in natural language mode` modifier, the `match ()` function performs a natural language search for a string against a text collection. For each row in the table, `MATCH ()` returns a relevance value, that is, a similarity measure between the search string and the text in that row in the columns named in the `match ()` list, (Widenius, Axmark, & AB).

Boolean searches are done using the `in boolean mode` modifier. With this modifier, certain characters have special meaning at the beginning or end of words in the search string. The `+` and `-` operators indicate that a word is required to be present or absent, respectively, for a match to occur.

Full Text Search also supports query expansion. This is generally useful when a search phrase is too short, which often means that the user is relying on implied knowledge that the full-text search engine lacks. For example, a user searching for `database` may really mean that `MySQL`, `Oracle`, `DB2`, and `RDBMS` all are phrases that should match `databases` and should be returned, too. This is implied knowledge.

The `LIKE` Transact-SQL predicate works on character patterns only, furthermore, string matching is usually limited to an exact match, or a wildcard match with the keyword "LIKE."

Wildcard and regular expressions require that MySQL try and match each and every row in a table, and table indexes are rarely used. As such, these searches can be very time consuming as the number of rows to be searched grows. Full text search on the other hand does not look at each row individually, analyzing and processing each word individually, rather when one sets up a full-text index on the product name and description field in a MySQL database, MySQL automatically stores the records in a indexed format, which means that MySQL can search the data in the fields very fast and efficiently in terms of server resources. Compared to the traditional like command in MySQL there is a lot performance to gain by switching to full-text search.

Full Text Searching also gives a relevance score which can be used to rank how best the text matches the pattern this is however not possible when using the like operator and the regular expressions.

In conclusion therefore, due to the advantages that fulltext search has over the other pattern

matching methods, this project used the fulltext search. Of the three modes of full text search, Boolean search is not applicable since it is not possible to know what to expect or what not to expect from the students, on the other hand blind query expansion tends to increase noise significantly by returning non-relevant documents, hence it is advisable to use it only when a search phrase is short, (Gusfield, 1997). As such fulltext search was used in natural language mode for this project.

2.3 INTELLIGENT TUTORING SYSTEMS MODELS

Two intelligent tutoring system architectures were very helpful in helping come up with a conceptual framework. The ITS architecture by Stankov and Glavinic (Stankov, Glavinic, & Rosic, 2000), supports a student-centered learning approach, it proposed 4 elements in the ITS, as depicted in figure 2.1.

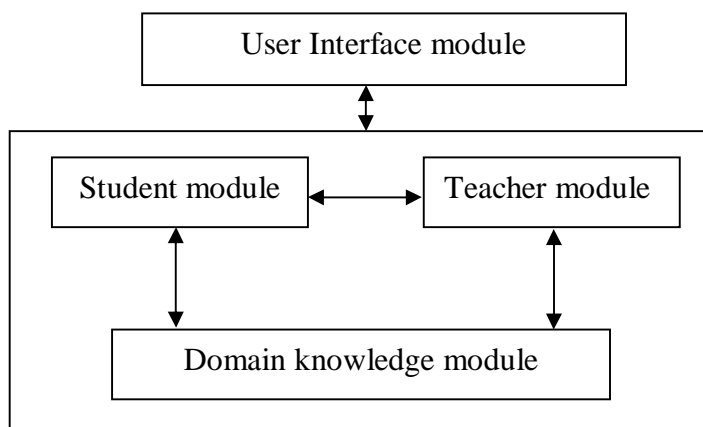


Figure 2.1 Architecture of ITS (Stankov, Glavinic, & Rosic, 2000)

2.3.1 Domain Knowledge Module

This module stores academic and related information

2.3.2 Teacher Module

This module deals with the teaching pattern for the learners. In case the learners want to revise the previous studied contents, the system presents new issues for learners, this element will store this information for helping the decision making of students afterwards.

2.3.3 Student Module

This stores the basic knowledge and fundamental information of the learners.

2.3.4 Interface Module

The interface module plays the roles of presentation and interaction between the users with ease of use.

The major advantage of this framework is that it is conceptually simple in that it can be categorized and studied through identifiable stages.

Another intelligent tutoring system architectures is shown in figure 2.2. According to the work of Stern and Haugsjaa (Woolf, 1992), there are 5 crucial elements in an ITS system;

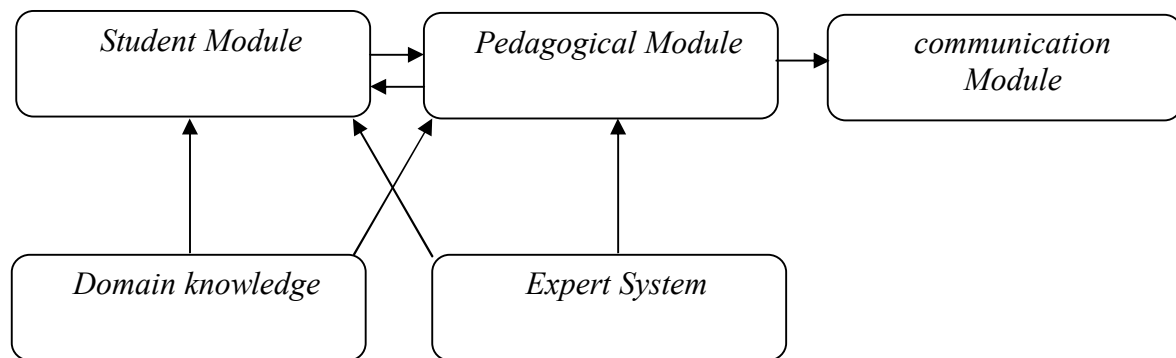


Figure 2.2 Architecture of ITS (Woolf, 1992)

2.3.5 Student Module

The student module stores information that is specific to each individual learner. The module tracks how well a student is performing on the material being taught. The purpose of the student module is to provide data for the pedagogical module of the system.

2.3.6 Domain Module

The domain module is the storage part for the information as the content, lesson, and teaching pattern. It works in the direction as, learning object, which metadata is the indicator for the characteristics of the content.

This component contains information the tutor is teaching, without it, there would be nothing to teach the student. The module requires significant knowledge engineering to represent a domain so that other parts of the tutor can access it.

2.3.7 Expert Module

The expert module is the storage part for rule base in solving problems for both the teacher and learners. It is similar to the domain knowledge in that it must contain the information being taught to the learner. However, it is also a model of how someone skilled in a particular domain represents the knowledge. By using an expert module, the tutor can compare the learner's solution to the expert's solution, pinpointing the places where the learner had difficulties.

2.3.8 Pedagogical module

This component provides a model of the teaching process. For example, information about when to review, when to present a new topic, and which topic to present is controlled by the pedagogical module. The student module is used as input to this component, so the pedagogical decisions reflect the differing needs of each student.

In other words this module gets information from student module and expert module and process the appropriate content and learning activities.

2.3.9 Communication module

Interactions with the learner, including the dialogue and the screen layouts, are controlled by this component. How should the material be presented to the student in the most effective way?

This module provides for the richness inherent in an intelligent tutoring system and is all-inclusive. It has however no interaction with the user. It provides information but has no room for the user to give such information. After studying these and other architectures it was possible to develop a conceptual model for the project.

2.4 THE CONCEPTUAL MODEL

The study of the existing architectures of intelligent tutoring systems, led to a conceptual model that has four main components as shown in figure 2.3, for the question analyzer.

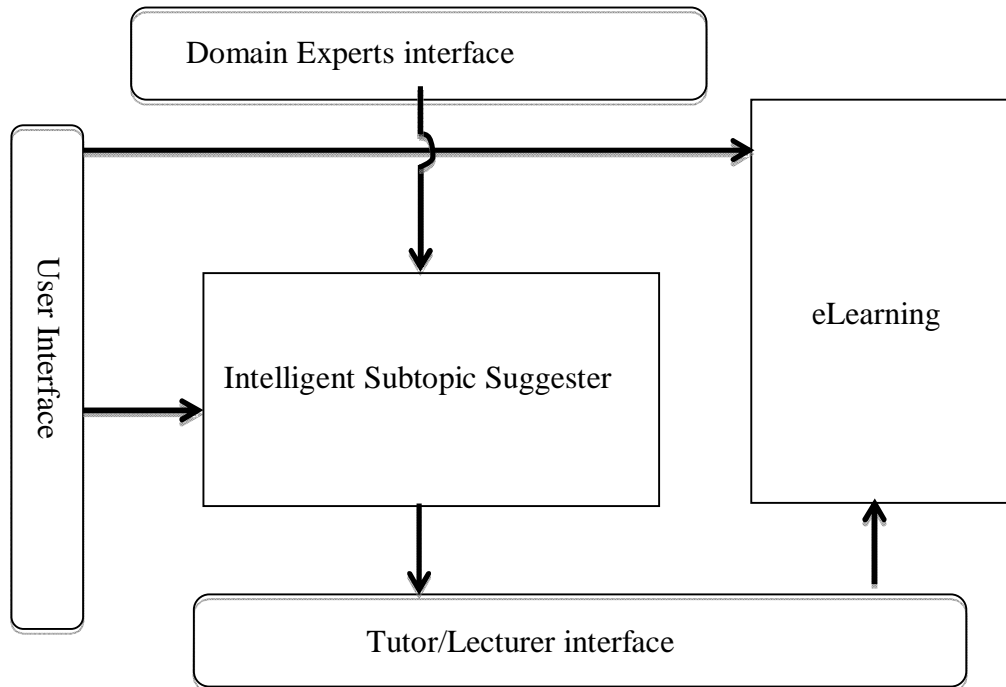


Figure 2.3: Conceptual model for the Intelligent Subtopic suggester

2.4.1 User Interface Module

This component consists of all screens of interaction between the users and the computer, it comprises the students, tutor/lecturer and DE interfaces. It controls communication between the learner and the system, the lecturer and the system, and then sends back information to the lecturer. The DE interface controls communication between the domain experts and the system.

2.4.2 Intelligent Subtopic Suggester

This component consists of all modules associated with the domain knowledge, the learners' questions, the pattern matching and ranking of subtopics. It is the repository of all relevant information, facts and knowledge acquired from the domain expert. It is also the storage for rules that control the working of the Intelligent subtopic suggester in solving the problem.

CHAPTER THREE METHODOLOGY

3.0 INTRODUCTION

Research methodology is a way to systematically solve the research problem. In this chapter we discuss the methods, tools and instruments used to achieve the objectives that were set in chapter 1.

3.1 RESEARCH METHODOLOGY

The project methodology involved defining the problem to be addressed, the general and specific objectives of the project were also established as stated in 1.3. A background research of the problem was then done, and this involved a literature review as described in chapter 2 to establish what is already known about the stated problem and to gather information to help in designing a conceptual model. In this project, prototyping methodology to design was largely followed to build and test the Intelligent Subtopic Suggester prototype.

The methodology also involved development of a prototype based on the developed model. The prototype was then evaluated to establish whether the prototype met the objectives of the project. Based on the results obtained a conclusion was made.

3.2 ISSUES THAT NEED TO BE ADDRESSED

The Product of chapter two was a conceptual model of the prototype to be designed as shown in figure 3.0. The interfaces that needed to be designed were quite clear from the model, however it was clear that the various components of the Intelligent Subtopic Suggester needed to be clearly identified.

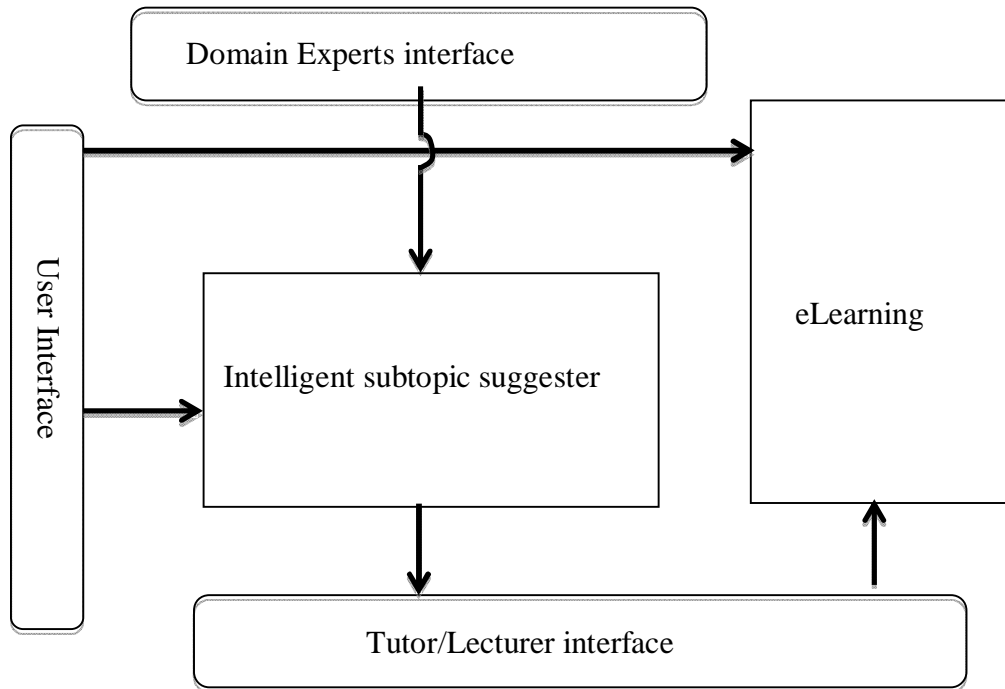


Figure 3.0: Conceptual model for the Intelligent Subtopic suggester

The design of the components of the Intelligent Subtopic Suggester was first done on paper before actually building anything, this helped to come up with the preliminary design that captured the best ideas of the project. The components of the suggester were identified as comprising; the pattern matcher, the question analyzer, the topic ranker and the knowledge base for the model as shown in figure 3.1.

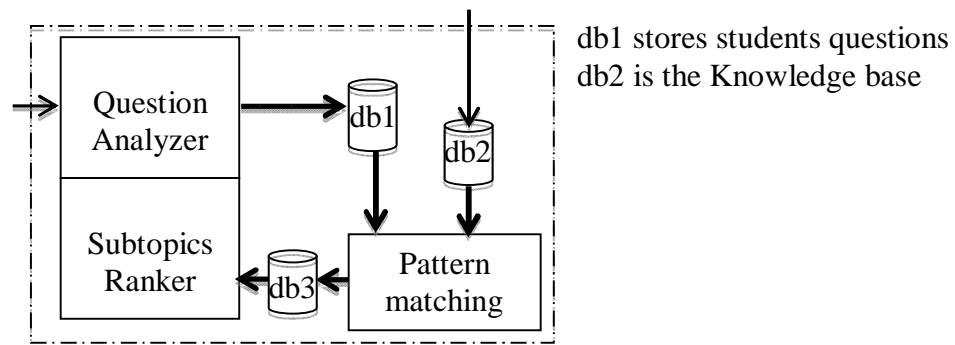
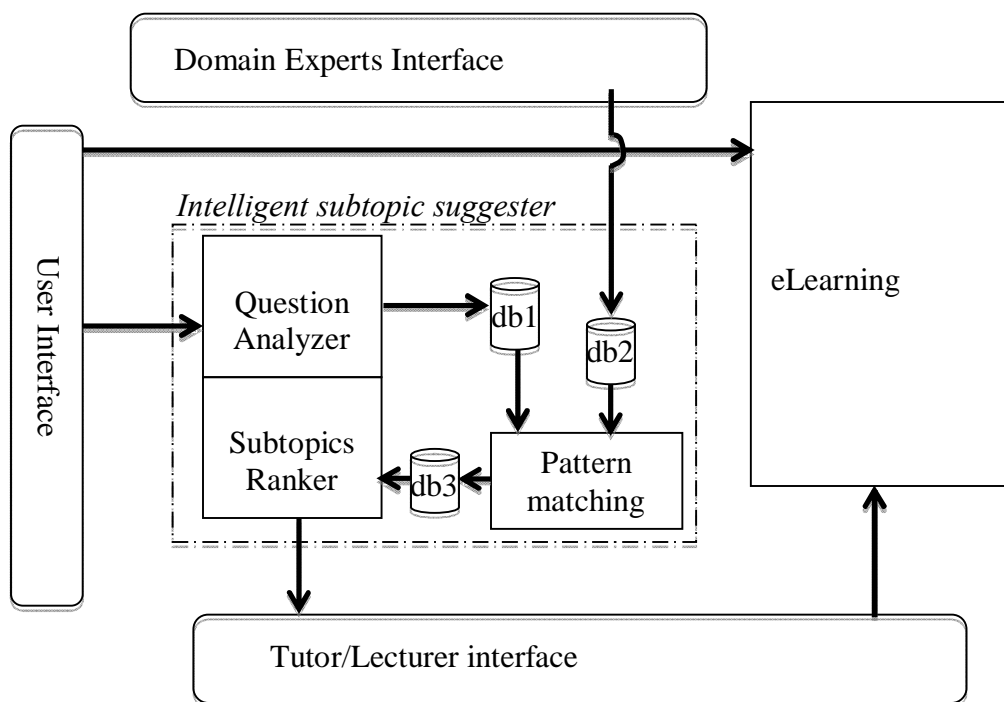


Figure 3.1: Components of the Intelligent Subtopic suggester

The Question analyzer handles all the functions of the student module that stores fundamental information of the learners including the students questions, in addition the suggester also has the domain knowledge that stores all the data collected from domain experts in the various disciplines of learning. It is the repository of all relevant information, facts and knowledge acquired from the domain expert. It is also the storage for rules that control the working of the Intelligent subtopic suggester in solving the problem.

Another component of the suggester is pattern matching, this is used to carry out inexact pattern matching between the terminologies in the students questions and the stored keywords. Within the Intelligent subtopic suggester, there is also the subtopic ranker that considers all the subtopics that match a question submitted by the students, and associates each question to a number of relevant subtopics and ranks the subtopics according to relevance respectively.

By incorporating these components of the suggester to the conceptual model, the architectural design of the suggester was developed as shown in figure 3.2.



db1 stores students questions; db2 is the Knowledge base

Figure 3.2: Architectural Design of the Intelligent Subtopic suggester

3.4 THE INTELLIGENT SUBTOPIC SUGGESTER PROCESS FLOW

To develop the intelligent subtopic suggester, various processes as shown in figure 3.3, were carried out.

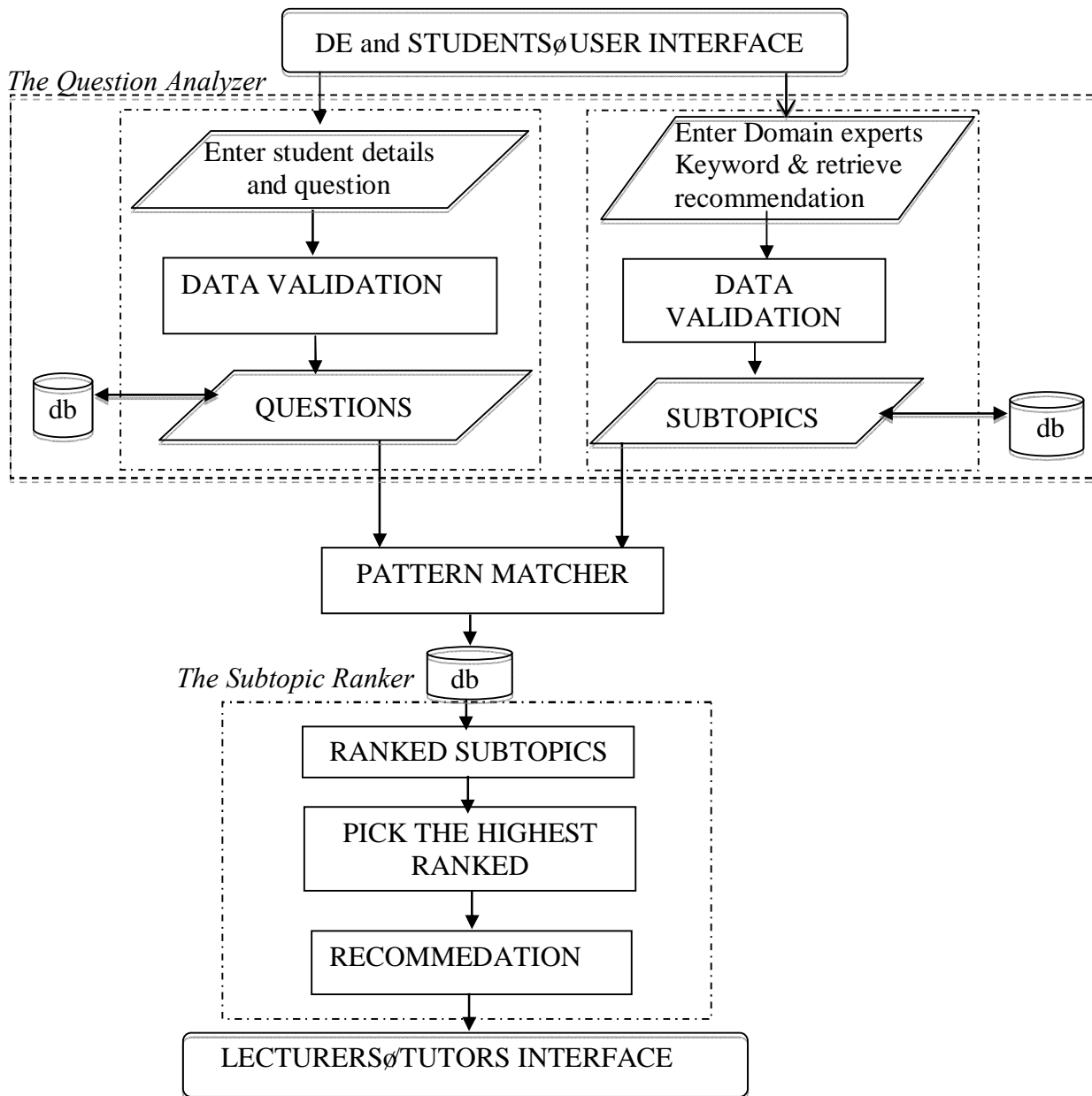


Figure 3.3 The Process Flow for the Intelligent Subtopic Suggester

3.5 DETAILED DESIGN OF THE COMPONENTS

3.5.1 Pattern Matcher Design

This module controls the working of the Intelligent Subtopic Suggester in solving the problem. The pattern matcher uses an inexact pattern matching algorithm which finds occurrences of bits of the pattern or the full pattern in this case the students questions, in the string in this case the subtopics and their associated keywords.

The pattern matcher was implemented using the MATCH () í .. AGAINST syntax of the Full-text searching with the IN NATURAL LANGUAGE MODE modifier, in this case the statement: `mysql_query ("SELECT * , MATCH (SubTopic) AGAINST ('$qns') AS relevance FROM Course_Outline WHERE MATCH (SubTopic) AGAINST ('$qns')");`

Helped to achieve the match. The *subtopics* refers to the data that had been stored by domain experts in db2, (figure 3.2) while *\$qns* is a variable that represents the students questions from db1.

Pattern matching each question does not uniquely identify a single subtopic of interest, instead several subtopics may match a particular questions. Several tests were run using the module to test if it was functioning as intended. The screen shot of Figure 3.4, is an example of one such output in which case the question was found to match several subtopics as shown;

6. Determine the separation distance between the plates of a parallel plate capacitor if the capacitance between them is and the enclosed area is 20cm²

Kingori	SSPI 1205	Technical Physics	CAPACITORS	FACTORS AFFECTING CAPACITANCE, dielectric constant, separation distance, area of plates, presence of dielectric
Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC FLUX strength of field passing through area, charge enclosed by surface area it passes through, orientation of area with respect to the field
Kingori	SSPI 1205	Technical Physics	CAPACITORS	CAPACITANCE measure of capacitor ability to store total charge, potential difference, separation distance, permittivity
Kingori	SSPI 1205	Technical Physics	CAPACITORS	TYPES OF CAPACITORS, variable and parallel plate capacitor, electrolytic, direct current between two sheets of aluminum foils, tantalum, paper soaked in wax or oil
Kingori	SSPI 1205	Technical Physics	CAPACITORS	ENERGY STORED, capacitance, charge of capacitor, potential difference, work done adding electrons to negative plate work done removing electrons
Kingori	SSPI 1205	Technical Physics	CAPACITORS	APPLICATIONS, Rectification, smoothing circuits, converting alternating current to direct current using diode, camera flash, tuning circuits has variable capacitor connected in parallel to inductor, delay circuits, Reduction of sparking in induction coil contact, filters Capacitor blocks direct current and allows alternating current
Kingori	SSPI 1205	Technical Physics	CAPACITORS	CAPACITORS IN PARALLEL ARRANGEMENT OR COMBINATION, total charge, voltage, capacitance
Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC POTENTIAL, force, test and point charge, uniformly charged disc, Equipotential surfaces, separation distance
Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	COULOMBS LAW, electric force between charges, separation distance, positive and negative point charges
Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC FIELD STRENGTH, field lines, separation distance, positive and negative point charges, force
Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC FIELD OR ELECTRIC FIELD INTENSITY, force, separation distance, positive and negative point charges

Figure 3.4: A screen shot of the data in the QA after pattern matching

3.5.2 Question Analyzer (QA) design

The QA was designed so that it was able to carry out various functions. To achieve this the QA has several sub modules. The QA ensures that no questions that have been analyzed are still stored in db3, it then checks if there are any questions that require to be matched, if there are none, it gives an appropriate report to the lecturer/tutor

The QA also preprocesses the questions by removing unacceptable symbols such as quotes and slashes that may have been used by students as they sent their questions. For example a question like; *'State Lenz's law'*, was changed to, *'State Lenzs law'*.

In case there are pending questions, the QA takes all the subtopics that match a particular question and stores them in db3, figure 3.2. In case the students question does not match any subtopic in the knowledge base, the QA was designed so that it gave a response that the question was unknown.

The QA was also designed so that it was able to store already analyzed questions in an archive.

The student module uses a form to capture a student's name, college number, email, course and question. When all the fields have been filled, the form was submitted using `post` method to a file that retrieves the values from the form and validates the values to ensure that no fields have been left unfilled.

Validation in this case also ensures that the student has entered a college number in the correct format, for example an acceptable format for this case would be P56/70746/07, while P/7078/2007 would be rejected. The email is also validated to ensure it follows an acceptable format. Validation is done using pattern matching with regular expressions.

Upon development of the complete module used to capture the students data, the module was tested to check whether it could detect a situation when some fields were left unfilled. It was also tested to ensure that it was able to do the validation as required, in this case an attempt was made to submit a form using a wrong format and the correct format. If it was found not to function as desired the module was redesigned and retested. This iteration was done severally and once the objectives were achieved the module was considered to be complete.

The module was also tested to ensure that it was able to insert data into db1. In case it was not possible the module was redesigned, the process of testing and redesigning was repeated until it was possible to insert the data correctly.

The QA also comprised the DE module. This module captured the lecturer's name, the code that describes the unit, for example SSPI 1206, the course being taught the unit and the main topic. In addition it collects information on the subtopics to be covered under a given unit. When all the fields have been filled, the data is submitted to a file that retrieves the values and validates the data. It ensures that the DE has filled in all the fields of the form. It also ensures that the name of the unit code is in the correct format, for example, SSPI 2201. Validation in this case also ensures that a semicolon, ; has been used to separate the various subtopics, because the ; marks the point along which various subtopics are split, so that all the parts of the line of text were then be put in an array. Once validation was done and the subtopics put into an array, the data was stored appropriately in db2.

Upon development of the complete module used to capture the DE's data, the module was tested to check whether it could detect a situation when some fields were left unfilled. It was also tested to ensure that it was able to do the validation as required. If it was found not to function as desired the module was redesigned and retested. This iteration was done severally and once the objectives were achieved the module was considered to be complete.

The module was also tested to ensure that it was able to insert data into db2, incase it was not possible to insert the data, the module was redesigned, the process of testing and redesigning was repeated until it was possible to insert the data correctly.

3.5.2.1 Student module Testing Results

The results of some of the tests run using the student module are given in table 3.1.

Table 3.1: student module Unit testing

FIELD	DATA SUBMITTED	OBSERVATION/SCREEN MESSAGE
Name:		Your mail could not be sent due to input errors
Col.No:		You forgot to enter your name!
E-Mail		You forgot to enter your college number!
Course:	Choose your course	You forgot to enter your email!
		You forgot to enter your college QUESTION!
Name:	Rose	Your mail could not be sent due to input errors Please enter a valid college number Enter a valid email
Col.No:	QP20/00247/07	
E-Mail	Rose.com	
Course:	Computer science	
Name:	Rose	Your mail could not be sent due to input errors Please enter a valid college number Enter a valid email
Col.No:	P56/0247/09	
E-Mail	rose@gmail	
Course:	Computer Science	
Name:	Rose	Thankyou for your question do you have another question
Col.No:	P56/00247/09	
E-Mail	rose@gmail.com	
Course:	Computer Science	

3.5.2.2 Result Interpretation and Conclusion

From the results of table 3.1, it is clear that the student module is able to detect invalid college numbers and email addresses that do not use the conventional format, that is it has the ability to reject the irregular input, the comments on the screen are also appropriate. The student module also detects blank fields and made the appropriate comments. When the data submitted is correct in terms of format the data is submitted and the user informed so. It was also noted that it was able to insert data into db1.

In conclusion the module was found to be operating as designed.

3.5.2.3 DE module Testing Results

The results of some of the tests run using the DE module are given in table 3.2.

Table 3.2: DE module testing

FIELD	DATA SUBMITTED	DATA STORED IN DB2				
LECTURER	Grace	Lecturer	UnitName	Course	Main topic	subtopic
Unit Code	SSPI 1205	Grace	SSPI	Technical	ALPHABETS	A
Course	Technical	Grace	SSPI	Physics	ALPHABETS	B
	Physics			Technical		
MAINTopic	ALPHABETS	Grace	SSPI	Physics	ALPHABETS	C
Subtopic	A : B : C:D			Technical		
		Physics	ALPHABETS	D		

3.5.2.4 Result Interpretation and Conclusion

From the results in table 3.2, it is clear that the data submitted using DE interface has been stored. It was also noted that the module was splitting the subtopics as required at the point where we have a semicolon. It was concluded that the DE module was working as designed.

3.5.3 Subtopic ranker Design

The subtopic ranker ranks the retrieved subtopics using a relevance measure that is, a number that tells us how good a match is. It uses the relevance rank formula:

$$w = \frac{(\log(dtf)+1)}{\text{sumdtf}} \cdot \frac{U}{(1+0.0115 \times U)} \cdot \log\left(\frac{(N-nf)}{nf}\right) [30].$$

Where:

- dtf is the number of times the term appears in the document
- sumdtf is the sum of (log(dtf)+1)'s for all terms in the same document
- N is the total number of documents
- nf is the number of documents that contain the term

U is the number of unique terms in the document

The formula has three parts: base part, normalization factor, global multiplier.

The base part is the left of the formula, $\frac{(\log(dtf)+1)}{sumdtf}$. The normalization factor is the middle part of the formula, $\left(\frac{U}{(1+0.0115 \times U)}\right)$. The idea of normalization is, if a document is shorter than average length then weight goes up, if it's average length then weight stays the same, if it's longer than average length then weight goes down. We're using a pivoted unique normalization factor, the word "unique" here means that our measure of document length is based on the unique terms in the document, 0.0115 is chosen as the pivot value.

When retrieving a text, rank is the product of the weight and the frequency of the word in the query: $R = (w \times qf)$

Where:

w is the weight

qf is the number of times the term appears in the query

R is a nonnegative floating-point numbers, which stands for Rank or Relevance

In conclusion weight, w , goes up if the term occurs more often in a row, goes down if the term occurs in many rows, goes up if the number of unique words in a row is fewer and goes down if the number of unique words in a row is more than average.

Therefore, a word that is present in many documents has a lower weight, because it has lower semantic value in this particular collection. Conversely, if the word is rare, it receives a higher weight. In order to obtain the most relevant subtopic from which a question has been obtained, the query tries to highlight certain words present in the question as significant words. Words referring full or part of the subtopic are considered significant.

Figure 3.5 is a screen shot showing the relevance of the returned result of one of the questions in particular the question "*Determine the separation distance between the plates of a parallel plate capacitor if the capacitance between them is 4×10^7 and the enclosed area is 20cm^2* " that was tested using the Intelligent Subtopic suggester.

```
mysql> SELECT *, MATCH (SubTopic) AGAINST (' Determine the separation distance between the plates of a parallel plate capacitor if the capacitance between them is 4*107 and the enclosed area is 20cm2 ') AS relevance FROM Course_Outline WHERE MATCH (SubTopic) AGAINST ('Determine the separation distance between the plates of a parallel plate capacitor if the capacitance between them is 4*107 and the enclosed area is 20cm2') HAVING RELEVANCE > 0.1ORDER BY RELEVANCE DESC;
```

id	LECTURER	UnitName	Course	MainTopic	SubTopic	relevance
20	Kingori	SSPI 1205	Technical Physics	CAPACITORS	FACTORS AFFECTING CAPACITANCE, dielectric constant, separation distance, area of plates, presence of dielectric	10.93124771118164
10	Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC FLUX strength of field passing through area, charge enclosed by surface area it passes through, orientation of area with respect to the field	7.340002059936523
15	Kingori	SSPI 1205	Technical Physics	CAPACITORS	CAPACITANCE measure of capacitor ability to store total charge, potential difference, separation distance, permittivity	7.1990966796875
19	Kingori	SSPI 1205	Technical Physics	CAPACITORS	TYPES OF CAPACITORS, variable and parallel plate capacitor, electrolytic, direct current between two sheets of aluminum foils, tantalum, paper soaked in wax or oil	6.836569309234619
18	Kingori	SSPI 1205	Technical Physics	CAPACITORS	ENERGY STORED, capacitance, charge of capacitor, potential difference, work done adding electrons to negative plate work done removing electrons	5.888632297515869
22	Kingori	SSPI 1205	Technical Physics	CAPACITORS	APPLICATIONS, Rectification, smoothing circuits, converting alternating current to direct current using diode, camera flash, tuning circuits has variable capacitor connect d in parallel to inductor, delay circuits, Reduction of sparking in induction coil contact, filters Capacitor blocks direct current and allows alternating current	4.050539970397949
17	Kingori	SSPI 1205	Technical Physics	CAPACITORS	CAPACITORS IN PARALLEL ARRANGEMENT OR COMBINATION, total charge, voltage, capacitance	3.943923234939575
14	Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC POTENTIAL, force, test and point charge, uniformly charged disc, Equipotential surfaces, separation distance	3.554717540740967
8	Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	COULOMBS LAW, electric force between charges, separation distance, positive and negative point charges	3.4381072521209717
13	Kingori	SSPI 1205	Technical Physics	ELECTRIC FIELD	ELECTRIC FIELD STRENGTH, field lines, separation distance, positive and negative point charges, force	3.412276029586792

Figure 3.5: A screen shot showing the relevance of the returned result.

From the screen shot we note that the subtopic “*FACTORS AFFECTING CAPACITANCE, dielectric constant, separation distance, area of plates, presence of dielectric*” matches with a relevance of 10.93124771118164 while the subtopic “*ELECTRIC FLUX strength of field passing through area, charge enclosed by surface area it passes through, orientation of area with respect to the field*” matches the same question with a relevance of 7.340002059936523, other subtopics also match but with much lower relevancies.

Once ranking has been done, the subtopic with the highest relevancy is picked as the subtopic from which the question has been asked, in the example above the subtopic associated with the question “*Determine the separation distance between the plates of a parallel plate capacitor if the capacitance between them is 4×10^7 and the enclosed area is 20cm^2* ” is “*FACTORS AFFECTING CAPACITANCE, dielectric constant, separation distance, area of plates, presence of dielectric*”.

Subtopics so picked are then counted to determine the frequency of questions from each subtopic. The recommendations of the Intelligent Subtopic Suggester is based on this frequency

3.5.4 Knowledge Base Design (db2)

This was designed to capture the knowledge necessary for the proper working of the Intelligent Subtopic Suggester, since it captures the expertise of the DE. The process of gathering the knowledge to stock the expert system's knowledge base involved extracting knowledge from domain experts (Physics lecturers in this case). This process of knowledge elicitation involved interview and non-interview techniques such as studying textbooks with the areas of interest, and other official records and documents such as moderation notes made during moderation of exams. An outline of the content to be covered and a questionnaire were used to guide the interview process. The research instrument used in this case was a questionnaire that used open-ended questions. The questions were unstructured in order not to restrict the respondents.

When eliciting knowledge from the DE, knowledge verification was done to help make quality assurance of the acquired knowledge. It involved review procedure, and multiple expert conflicts resolving procedure.

Reviewing at the elicitation stage was conducted by letting the domain experts review the results of the knowledge elicitation sessions. At the analysis and modeling stage, the domain experts reviewed the filled forms describing the domain knowledge. Reviewing at the implementation stage was conducted by letting the domain experts review the early prototype.

In multiple experts conflict resolution, when two experts gave different knowledge for the same thing, an effort was made to resolve this conflict in order to store knowledge agreed upon by both of them. If no consensus was reached, the expert who is recognized to be more specialized in the area of disagreement was considered.

Once the knowledge elicitation stage was over, knowledge representation that involved the process of translating data extracted from the problem domain experts into the data structure of the knowledge base was done. The data extracted was in this case organized in a way that could be stored using MySQL tables.

3.6 PROTOTYPE DEVELOPMENT APPROACH

An evolutionary prototype was used for the project, in this case the initial prototype was expanded by adding more and more knowledge and functions, until it became the final prototype. The steps followed in the prototype development were in line with the prototyping process of figure 3.6.

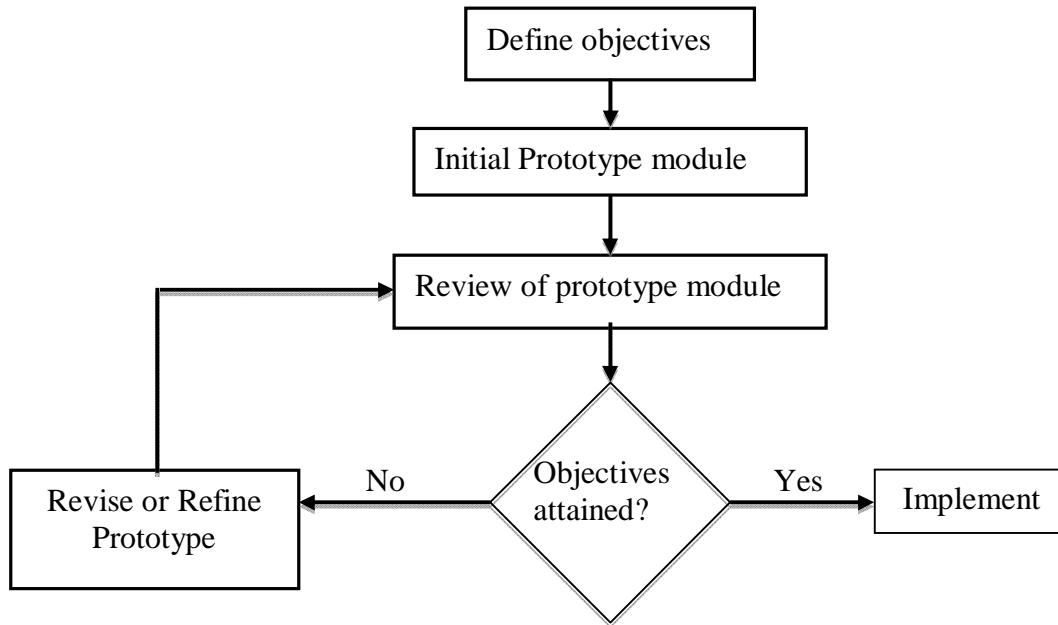


Figure 3.6: The Prototyping Process

3.6.1 SELECTION OF THE PROTOTYPE’S DEVELOPMENT TOOLS

This involved selection of the tools to be used to design the Intelligent Subtopic Suggester, this required both hardware and software. The software required was a web server/hosting that supports PHP and MySQL databases, a text editor such as notepad, and Microsoft word. PHP 5.3.5 was used for scripting system functionalities, because it is compatible with MySQL and allows effective connectivity and interactivity to the database. Its syntax and semantics are learnable and efficient.

Hardware requirement for this project will be hard disk space to store the questions and subtopics keywords.

3.6.2 Database design

MySQL client version 5.5.9 provided the database capability for storing and managing details of the users. MySQL is a relational database management system (RDBMS) and uses tables to represent information. Various tables were created to store data for the various modules. Some of the tables created for the Intelligent Subtopic suggester had the following fields

Table 3.3: Courseoutline table

Field name	Data type
id	int
LECTURER	Tinytext
UnitName	Varchar
Course	Tinytext
Maintopic	Varchar
SuBTopic	Text

Another table that was created stored the results of the question analyzer and had the following fields

Table 3.4 Analyzer Table

Id	Int
LECTUREREs	Tinytext
UNIT_Name	Varchar
Course	Varchar
MAIN_TOPIC	Varchar
SUB_TOPIC	Text

Another table StudentRecord table stored the student data

Table 3.5: StudentRecord table

Id	Int
RegNo	Varchar
Contact	Varchar
Course	Tinytext
Question	Text
SubmissionDate	Date

3.6.3 The User Interface Design

From the conceptual model it was clear that we needed to design the student, lecturer/tutor and domain experts interfaces. The user interface was designed in a modular form in order to capture data involving each user. This involved designing the data capture forms and the data presentation forms, for each of the users.

3.6.3.1 The Student User Interface

The student form captures basic minimal information about the student interested in using the system. The student registers using his or her names, and college number then selects the course they are taking from a drop down menu. Figure 3.7 is a screen shot of the students' interface.

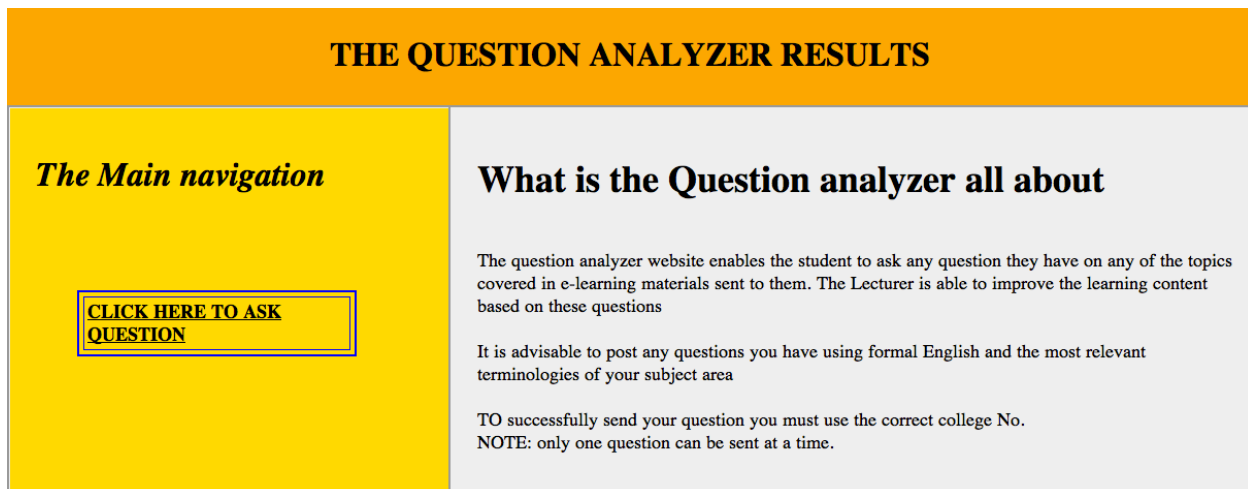


Figure 3.7: The student interface

3.6.3.2 The Domain Experts Interface

The domain expert's data was captured using a form. The domain expert registers using his or her names, the code that describes the unit, the course being taught the unit and the main topic. In addition the form also collects information on the subtopics to be covered under a given main topic. The DE module was tested to check whether it could detect a situation when some fields were left unfilled. It was also tested to ensure that it was able to do the validation as required. If it was found not to function as desired the module was redesigned and retested. This iteration was done severally and once the objectives were achieved the module was considered to be complete.

The module was also tested to ensure that it could insert data into db2, In case it was not possible to insert the data, the module was redesigned, the process of testing and redesigning was repeated until it was possible to insert the data correctly. Figure 3.8 is a screen shot of the domain experts' interface.

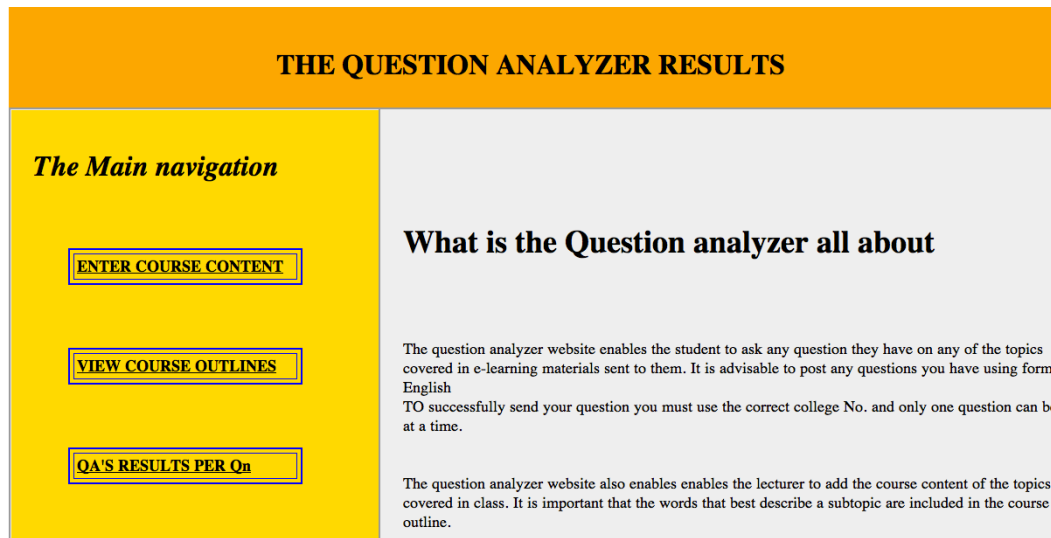


Figure 3.8: A screen shot of the domain experts' interface.

3.2.3.3 The Lecturers Interface

This was designed using PHP scripting and it allowed the lecturer to view the students questions that await to be analyzed, the archive questions and the results of the Intelligent subtopic Suggester. Figure 3.9 is a screen shot of the lecturers' interface.

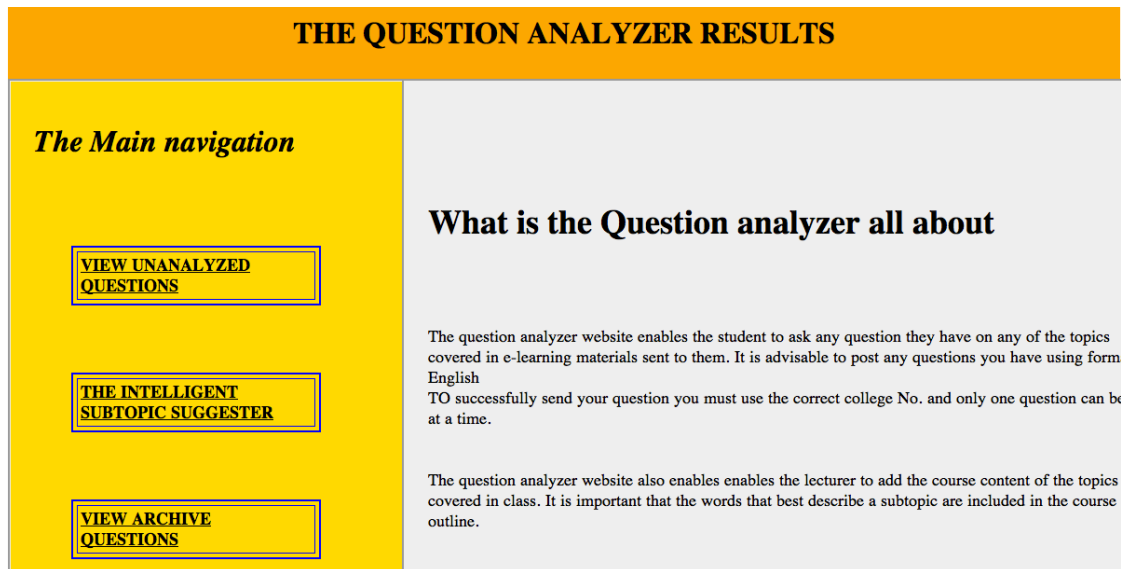


Figure 3.9: A screen shot of the lecturers' interface.

3.7 IMPLEMENTATION OF THE MODULES

The physical design specifications were turned into working computer code using PHP scripting. Different modules of the prototype were developed separately as described, each module was then tested for any unanticipated flaws, things that may have been overlooked or design features that did not work as intended. The modules were then combined to form the Intelligent Subtopic Suggester prototype.

Implementation of the Intelligent Subtopic Suggester used production rules such as;

```
// Retrieve all the data from the "StudentRecord" table
$result = mysql_query("SELECT * FROM StudentRecord") or die(mysql_error());
$numrows = mysql_num_rows($result);

    if( $numrows < 1)

    { echo "<p align='center'><font color=blue size='7pt'><br><br><b>SORRY.<br>
    THERE ARE NO QUESTIONS REQUIRING YOUR ATTENTION</b></font></p>"; }
    while($row= mysql_fetch_array($result))
    { $myqn=$row['Question'].":";
    $myqns=explode(":", $myqn);
```

```

        foreach($myqns as $qns)
    { //start foreach

        if ($qns!="")

        {$query = "SELECT * , MATCH (SubTopic) AGAINST ('$qns' ) AS relevance FROM
        Course_Outline WHERE MATCH (SubTopic)  AGAINST ('$qns')  ORDER BY
        RELEVANCE DESC LIMIT 1";

        $numresults = mysql_query ($query)or die(mysql_error());
        $ReturnedValue= mysql_num_rows($numresults);

        if ($ReturnedValue<1)

        { $query = "SELECT * FROM Course_Outline WHERE SubTopic LIKE '%$qns%'
        LIMIT 1";

        $numresults = mysql_query ($query) or die ( "Couldn't execute query" );
        $ReturnedValue1 = count ($numresults); }
    }

```

These and other rules that can be seen in the sample code (Appendix A) helped to implement the Intelligent Subtopic Suggester. Once implementation was completed experimental tests were run using training data that had been collected from course textbooks. The aim of the tests was to verify if all the modules were working as expected after they were combined.

The training data gave results that had a precision of 0.890, a recall of 1 and an accuracy of 0.810. Based on these results it was concluded that the Intelligent Subtopic Suggester was functioning as planned.

CHAPTER FOUR PROTOTYPE EVALUATION

4.0 INTRODUCTION

The main purpose of evaluation is to make conclusions about the developed prototype and to improve its effectiveness. In chapter 3 various tests were run to test the working of individual component to ensure that each functions properly as a unit, these tests showed that the various units were functioning as desired.

Integration testing was also done after the unit testing to verify that all elements of the prototype mesh properly and that the overall system function/performance had been achieved. From the results obtained it was noted that the design objectives of the project had been achieved since for every input the QA was giving an output that correspond to the subtopic from which the question had been derived.

A comparison of the current eLearning systems with an eLearning system that has the QA module shows that the eLearning system with a QA gives feedback to the course tutor. This is an achievement for the eLearning system, since based on the subtopic from which most questions are asked by the student, the lecturer can review the content that has been provided to the student through the various technologies. Such a review may involve giving additional content to the students are using a different approach.

When all these tests had been done, it was now time to do the prototype evaluation. It was used for overall testing of the prototype after the end of the development process, to verify that the developed prototype meets requirements or to identify differences between expected and actual results. Black-box testing approach in which test data are derived from the specified functional requirements without regard to the final program structure were used exclusively during evaluation.

Evaluation was done to measure how well or badly the prototype performs on some new, previously unseen data. Evaluation also makes it possible to compare prototypes built for the same task.

In order to carry out reliable evaluation of the prototype, several rules were followed. One was to separate the training and test data sets. The training set (seen data) was used to build the prototype and the test set (unseen data) was used to measure the prototype's performance, at this point no changes were done to the knowledge base or any modules of the prototype. This made it possible to obtain predictive accuracy figures for the prototype, and helped to avoid over fitting

on a particular test data set.

The Evaluation process was performed by subjecting the Suggester to two sets of test cases. The test cases were solved by both the Intelligent Subtopic Suggester and the domain expert. The basic idea of the adopted technique was to evaluate the behavior of the Expert system, against that of human experts. Human experts in the domain, as well as the expert system, solve these test cases. Each human expert works out the test cases independently. Solutions of test cases were then evaluated in a blind manner, so that distinguishing between solutions of the expert system and solutions of domain experts became impossible.

Domain experts, other than those who gave the knowledge acquired by the expert system and other than those who solved the test cases were then given test case solutions for evaluation. Later on, an open discussion was held to let the human experts justify their solutions.

Standards for evaluation metrics were used to evaluate the performance of the intelligent subtopic suggester prototype. One of the standards, precision is a measure of the correctness of the prototype's output to that of the manual system to classify the same questions, that is how much of the information that the system returned is actually correct, this was defined as shown in equation 4.1.

$$Precision = \frac{TP}{TP + FP} \dots \dots \dots (4.1)$$

The other standard used to evaluate the prototype is recall, which measures how much relevant information is extracted, this was defined as shown in equation 4.2

$$Recall = \frac{TP}{TP + FN} \dots \dots \dots (4.2)$$

Accuracy was also another performance measure used to evaluate the prototype, accuracy refers to the proximity of measurement results to the true value, this was defined as shown in equation 4.3

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots \dots (4.3)$$

Where

True positive(t_p) refers to the number of questions correctly labeled as belonging to a particular

subtopic, that is correctly identified.

False positive (fp) refers to the number of questions incorrectly labeled as belonging to a particular subtopic, that is incorrectly identified questions.

False negative (fn), refers to the number of questions that were not labeled as belonging to a particular subtopic but should have been, that is incorrectly rejected.

True negative (tn) refers to the number of questions that were not labeled as belonging to a particular subtopic and should not have been labeled, that is correctly rejected, (Poers & M, 2011).

4.1 DATA COLLECTION FOR THE TEST

Test data was data that had not been used during the process of developing the prototype. It was obtained from three main sources; some data was collected from students using the Intelligent Subtopic Suggester prototype, another source was Internet blogs and questions from domain experts, in this case a standard exam past paper in the relevant area was randomly picked.

The most import criteria of these test cases are that they cover normal cases, as well as the most standard, and rare cases.

4.2 CLEANING THE DATA

Data cleaning was done to make a data set consistent with other similar data sets in the system. Once the data for testing was collected the data that was used in purpose sampling was cleaned. Cleaning involved elimination through deletion of incomplete questions and removal of typographical errors, ambiguous questions were also eliminated.

Data cleaning also involved harmonization of data, this involved harmonization of short codes for example *pd* to actual words *potential difference* in this case.

4.3 DATA SAMPLING

Data from past papers was divided into two, one part of the questions was tested in its raw form while the second part had purpose sampling employed on the questions so that the test questions chosen were questions that were not referring to any diagrams.

For the questions obtained from Internet blogs on topics of interest, the test data was classified into two

- (i) Some questions were tested in their raw form without eliminating any questions
- (ii) Purpose sampling was employed so that the test questions were questions that are academic. In this case questions like those asking where to purchase or where certain items are manufactured were eliminated from the test. Ambiguous questions were also eliminated, for example a question "Draw and label a parallel and series circuit" was one of those eliminated since it is not clear whether it is a circuit involving resistors, capacitors or inductors.

4.4 MODE OF ANALYSIS

The mode of analysis is deductive based on descriptive statistics method. The performance standards will be considered for the various sets and comparisons then made between the accuracy of placement based on data from the various sources.

4.5 RESULTS

The following results were obtained using the Intelligent Subtopic Suggester. The results were computed in line with the performance standards to be used

4.5.1 Summary of the results of the data from Internet

The data from the Internet had been divided into two sets, the results that compared the results of the performance of the prototype on data obtained from the Internet were as shown in table 4.1.

Table 4.1: Results comparing performance of the QA when used with questions from internet

	Questions from internet in their raw form	Sampled Questions from internet
Total Number of questions	52	52
True Positives	21	31
False Positives	28	21
False Negatives	1	0
True Negatives	2	0

4.5.2 Summary of the Results of the Data from Students and DE

The results that compared the results of the performance in identifying placement of questions submitted by students and those set by domain experts were as shown in table 3.4

Table 4.2: Performance of the questions from the students and domain experts table

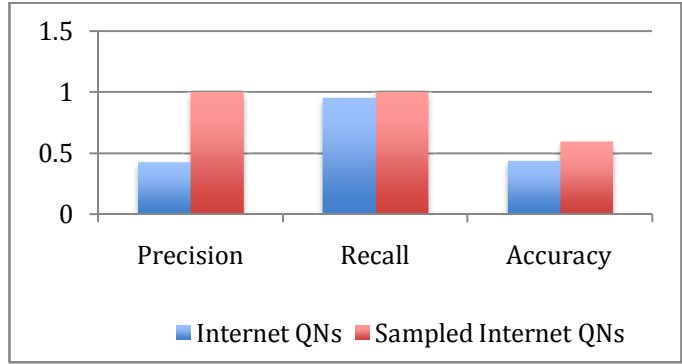
	Questions sent in by students studying the E & M unit	Questions from a randomly picked past paper	Sampled Questions from a randomly picked past paper
Total Number of questions	30	30	30
True Positives	22	18	23
False Positives	5	12	7
False Negatives	2	0	0
True Negatives	1	0	0

4.6 RESULTS ANALYSIS

From the results of table 4.1 and 4.2 various performance standards were calculated using equations 4.1, 4.2 and 4.3. The results of the calculations are shown in table 4.3 and 4.4

Table 4.3: Performance standards of the Intelligent Subtopic Suggester on Internet data

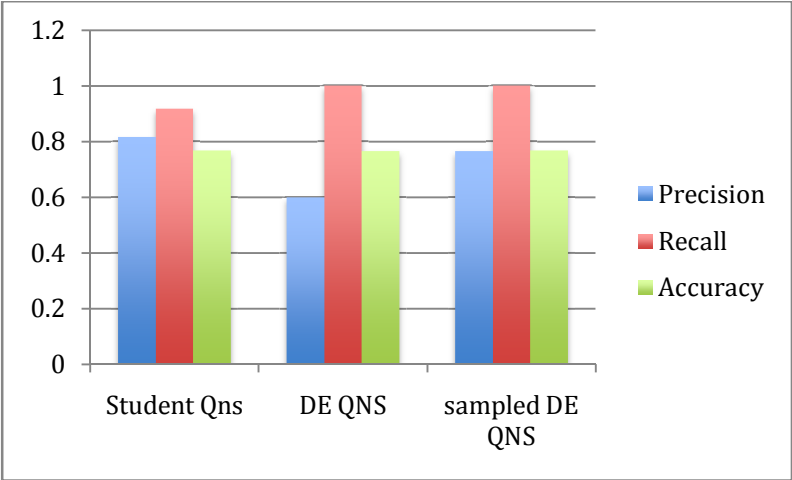
	Questions from internet in their raw form	Sampled Questions from internet
Precision	0.429	1
Recall	0.955	1
Accuracy	0.442	0.596



Graph 4.1: Performance standards of the Intelligent Subtopic Suggester on Internet data

Table 4.4: Performance standards of the Intelligent Subtopic Suggester on students and domain experts data

	Questions sent in by students studying the E & M unit	Questions from a randomly picked past paper	Sampled Questions from a randomly picked past paper
Precision	0.815	0.600	0.766
Recall	0.917	1	1
Accuracy	0.767	0.766	0.767



Graph 4.2: Performance standards of the Intelligent Subtopic Suggester on students and DE data

4.7 DISCUSSION OF RESULTS

In information retrieval, a precision score of 1.0 means that every result retrieved by a search was relevant, but says nothing about whether all relevant documents were retrieved, whereas a recall score of 1.0 means that all relevant documents were retrieved by the search, but says nothing about how many irrelevant documents were also retrieved. The precision of a measurement system, related to reproducibility and repeatability, is the degree to which repeated measurements under unchanged conditions show the same results. Accuracy is the overall correctness of the prototype. A measurement system is considered valid if it is both accurate and precise.

The quality of searching the right information accurately is the precision value of the Intelligent Subtopic Suggester, (Sampath & Pavithra, 2010). We note from the results of table 4.3 that for the Internet raw data, the Intelligent Subtopic Suggester was observed to retrieve a greater number of subtopics that were incorrectly labeled as belonging to a particular subtopic, as compared to those that had been sampled. This was attributed to the fact that some questions were ambiguous, while others were quite irrelevant, however, the fact that these questions had words that matched some of the words in the course outline meant that they ended up being matched to a subtopic.

The questions from the Internet were also not using the scientific terminologies, it was a case of trying to think what the user was asking. The fact that the questions were not clear was attributed to have caused the classification accuracy of around 60% even for the sampled data. Both sets had a recall of approximately 100%, which means that all relevant documents were retrieved by the search

The Intelligent Subtopic Suggester performed sufficiently well with the sampled questions from a past paper and the students' questions. The accuracy in all the cases was found to be 77%. The questions from a randomly picked past paper are well set, lack ambiguity and within the syllabus, however it was noted that the Intelligent Subtopic Suggester performed best in classifying questions sent in by the students and in the sampled questions than in those of the past paper.

This was attributed to the fact that most of the questions sent in by students were seeking clarifications of content that had already been covered. As such the students tended to use

descriptive terms that easily matched words stored in the knowledge domain. Furthermore the questions generally had few mathematical expressions or diagrams.

In conclusion it was noted that the question analyzer was able to identify the subtopic from which a question had been asked and performed well in those questions. However the question analyzer lacks the ability of pattern recognition and as a result performed poorly in the questions that were referring to diagrams and those using mathematical symbols.

CHAPTER FIVE CONCLUSION AND RECOMMENDATIONS

5.0 CONCLUSION

In Chapter 1 we explained that one of the major shortcomings of eLearning in Kenya is the fact that the eLearning platforms are mainly used to disseminate information. They provide a way for student to access learning content. However considering the number of students involved it is practically impossible for a lecturer to give individualized attention to a students' question in order to understand where they could be having problems with the content he delivers through the various technologies, consequently the lecturer usually has no way of knowing how effective any material delivered in such a manner is. The greatest achievement of this project is that it has made it possible to automate the process of going through the questions and identifying subtopics from which they have been asked.

The objectives that had been set at the beginning of the project were achieved. During the project a prototype of an Intelligent Subtopic Suggester was developed and tested. From the evaluators' results, we can conclude that the question analyzer can

- (i) Identify the subtopic from which a question has been asked, and hence can identify the students' key learning needs.
- (ii) Suggest to the tutors/lecturers subtopics that students may need further help with based on the frequency of question from the said subtopic

The highest precision for the Intelligent Subtopic Suggester was highest for the sampled questions from the Internet which had a value of 1, while the least was recorded for the questions from the internet in their raw form and had a value of 0.429. The recall value for the internet data was generally high ranging between 0.955 and 1, the accuracy however was 0.442 and 0.596 for the questions from the internet in their raw form and for the sampled questions from the internet respectively.

It was noted that except for internet data the Intelligent Subtopic Suggester had a precision measurements ranging between 0.60 and 0.82, since precision of a measurement system, related to reproducibility and repeatability, is the degree to which repeated measurements under unchanged conditions show the same results. We can conclude that the Intelligent Subtopic Suggester consistently placed the same question under the same subtopic and hence we conclude that the system is consistent and therefore reliable.

For questions asked by people familiar with the subject content, that is the domain experts and

students taking the course, the suggester averaged an accuracy of 0.766, a precision of 0.727 and a recall of 0.972 on average. A comparison of these results with other results of tests that have been done on search engines shows an acceptable comparison, specifically mean precision of Google is 0.800, of Yahoo is 0.750 metacrawler is 0.830 and of Dogpile is 0.880, while mean precision for Google is 0.920, of Yahoo is 0.070 metacrawler is 0.0.500 and of Dogpile is 0.490 (Sampath & Pavithra, 2010).

Another study by Geyer-Schulz and Hahsler an evaluation of recommender algorithms shows that at reasonable recall (between 20% and 50%) both algorithms reach a precision between 60% and 80% and an accuracy above 70% 35 (Andreas & Michael, 2002).

The performance standards of the suggester are therefore comparable with studies performed on similar systems, we can therefore conclude that the Intelligent Subtopic Suggester has made it possible to automate the process of analyzing a particular question and, to associate it with a particular subtopic. The Intelligent Subtopic Suggester is also able to get the frequency of questions from each subtopic and make a recommendation based on the frequency.

For both questions from internet in their raw form and questions from a randomly picked past paper the recall was high and precision low, this could mean that the prototype gave many possible answers at the cost of also giving wrong ones.

5.2 LIMITATIONS OF THE STUDY

During the study there were some challenges that were encountered. They included:

- ❖ Getting domain experts who were willing to devote time to be involved in the development of the Intelligent Subtopic Suggester. As a result the development of the prototype and its testing took longer than earlier planned.
- ❖ Getting data for testing the prototype was also a challenge since the student group taking the unit of interest had few students.

5.3 RECOMMENDATIONS

It would be recommended to pilot the application in various colleges, for students taking the electricity and magnetism unit, and maintenance of the Intelligent Subtopic Suggester to continue until optimal results are obtained using it. It is also recommended that any questions asked using the question analyzer should be very clear and should not be ambiguous in any way.

From the performance witnessed from questions by students who had actually studied the course it is advisable that the students using the question analyzer should be familiar with terminologies of the subject area.

5.4 SUGGESTIONS FOR FURTHER RESEARCH

To further widen and improve some of the findings of this study, further work needs to be carried out, the Intelligent Subtopic Suggester was observed to perform poorly in questions that had diagrams and mathematical symbols. It would be recommended that further research that would lead to incorporation of pattern recognition to the question analyzer be done in order to improve its ability to identify such patterns.

Further research needs to be done with domain experts in other subject/unit areas with an aim to expand the knowledge base, so that it addresses all the subject areas covered using eLearning in colleges

REFERENCES

- Adjeroh, D. A., Mukherjee, A., & Powell, M. (1997). Searching BWT Compressed Text with the Boyer-Moore Algorithm and Binary Search. *Journal of Visual Communication and Image Representation* , 8 (2), 182-207.
- Alkhattabi, M., Neagu, D., & Cullen, A. (2010). Information Quality Framework for ELearning Systems . *An International Journal (KM&EL)* , 2 (4), 340-362.
- Alqadi, Z. A., Aqel, M., & El Emary, I. M. (2007). Multiple Skip Multiple Pattern Matching Algorithm (MSMPMA). *International Journal of Computer Science* , 34 (2).
- Andreas, G.-S., & Michael, H. (2002). Evaluation of Recommender Algorithms for an Internet information broker Based on Simple Association Rules and on the Repeat Buying Theory. *Forth WEBKDD'2002 Workshop*, (pp. 100-114). Edmonton Canada.
- Bell, T., Matt, P., Mukherjee, A., & Adjeroh, D. (2002). Searching BWT Compressed Text with the Boyer-Moore Algorithm and Binary Search. *DCC '02 Proceedings of the Data Compression Conference* (p. 112). Washington, DC, USA: ACL.
- Eliot, C., & Woolf, B. (1995). An Adaptive Student Centered Curriculum for an Intelligent Training System. *User Modelling and User adapted interaction* , 5 (1), 67-86.
- Feldman, S. (1999, May). NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval. *BIOC 218* .
- Gusfield, D. (1997). *Computer Science and Computational Biology*. United Kingdom: Cambridge University Press.
- Gyorodi, R. C., Gyorodi, R., Pecherle, G. M., Gyorodi, C. C., & George, M. (2010). Full-Text Search Engine using MySQL. *International Journal of Computers Communications & Control* , 5 (5), 735-743.
- Hafner, K. (2004, September 2004). Software Tutors Offer Help and Customized Hints. *New York Times* .
- Khan, Z. A., & Pateriya, R. K. (2012). Multiple Pattern String Matching Methodologies: A Comparative Analysis. *International Journal of Scientific and Research Publications* , 2 (7).
- Lajoie, S., & A, L. (1992). In Intelligent Instruction by Computer. In F. a. Psotka, *Apprenticeship Training in the Workplace: Computer-Coached Practice Environment as a New Form of Apprenticeship* (pp. 15-36). Washington DC: Taylor & Francis.

- Lee, J. (2004, June 4). BIOC 218. *Analysis of a fundamental exact and Inexact pattern matching Algorithms* .
- Mohammad, A., Saleh, O., & Abdeen, R. A. (2006). Occurrences Algorithm for String Searching Based on Brute-force Algorithm . *Journal of Computer Science* , 2 (1), 82-85.
- Odhiambo, & Omondi, O. O. (2009). *Union List of Theses and Dessertation Reflecting Research on Kenya*. Retrieved November 14, 2013, from researchkenya.or.ke:
:http://researchkenya.or.ke/node/16588
- Omwenga, E., & Rodrigues, A. (2006). Towards an Education Evaluation Framework: Synchronous and Asynchronous E-Learning. *Journal of the Research Center for Educational Technology (RCET)* , 2 (1), 46-59.
- Ouma, M. (2013, November 17). *Intel's "Explore and Learn" platform to enhance e-Learning in Kenya, Nigeria and South Africa*. Retrieved 3 23, 2014, from aptantech simply about ICT and YOU: <http://aptantech.com/2013/11/intels-explore-and-learn-platform-to-enhance-e-learning-in-kenya-nigeria-and-south-africa/>
- Poers, W., & M, D. (2011). Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies* , 2 (1), 37-63.
- Pollack, R., & Ute, S. (2007, November 5). An Intelligent E-Learning System for Beginner Programming - Using Analogical Reminder for Error Classi. *Master Thesis* .
- Sampath, K. B., & Pavithra, S. M. (2010). Evaluating the searching capabilities of search engines and metasearch engines: a comperative study. *Annals of library and information studies* , 57, 87-97.
- Shute, V., Glaser, R., & Raghaven, K. (1989). Inference and Discovery in an Exploratory Laboratory. *Learning and Individual Differences* , 279-326.
- Singh, M. (2003). Analysis of Karp Rabin Algorithm. *Int J Bioinform Res Appl.* , 6 (2), 147-162.
- Singla, N., & Garg, D. (2012). String Matching Algorithms and their Applicability in various Applications. *International Journal of Soft Computing and Engineering (IJSCE)* , 1 (6), 2231-2307.
- Ssekakubo, G., Suleman, H., & Marsden, G. (2011). Issues of adoption: have e-learning management systems fulfilled their potential in developing countries? *SAICSIT Annual Research Conference 2011* (pp. 231-238). capetown: ACL.

- Stankov, S., Glavinic, V., & Rosic, M. (2000). On knowledge Representation in an Intelligent Tutoring System. *Proc. 4th IEEE International Conference on Intelligent Engineering Systems* , 17-19.
- Stephen, A. G. (n.d.). *Searching Algorithms*. Retrieved October 10, 2013, from Softpanorama: <http://www.softpanorama.org/Algorithms/searching.shtml>.
- Strzalkowski, T., & Vauthey, B. (1992). Information Retrieval using Robust Natural Language Processing. *30th Annual Meeting on Association for Computational Linguistics* (pp. 104 -111). Stroudsburg, PA, USA: ACL.
- Tiwari, A. R., & N, K. S. (2012). String Matching Methodologies:A Comparative. (*IJCSIT*) *International Journal of Computer Science and Information Technologies* , 3 (2), 3394 - 3397.
- Veeramani, M. (2010). E-Learning: A Conceptual Framework. *International Journal of Educational Research and Technology* , 1 (2), 20-24.
- Widenius, M., Axmark, D., & AB, M. (n.d.). *MySQL 5.5 Reference Manual*. Retrieved 4 4, 2014, from MySQL: <http://dev.mysql.com/doc/refman/5.5/en/>
- Woolf, B. (1992). AI in Education. In S. Shapiro, *Encyclopedia of Artificial Intelligence* (pp. 434-444). New York: John Wiley & Sons.

APPENDIX A: sample code

Student data form

```
<html>
<body>
    <form action="StudentFormFinale.php" method="post" style="padding: 8px; margin:
0px; background: gold;">
        <fieldset>
<p align='center'><font color=blue size='5pt'><b> PERSONAL DETAILS</b></font></p>
        <tablecellspacing="9" cellpadding="2" border="2" width="100%" >
            <tr><td align="right">Name:</td><td><input type="text" name="fname" value=""
size="25" maxlength="50" /></td></tr>
            <tr><td align="right">Col.No:</td><td><input type="text" name="col_no" value=""
size="15" maxlength="15" /></td></tr>
            <tr><td align="right">EMail:</td><td><input type="text" name="EMail" value=""
size="25" maxlength="50"/></td></tr>
            <tr><td align="right">Course:</td><td><select name="Course">
                <option value="1" selected>Choose your Course</option>
                <option value="2">computer science</option>
                <option value="3">Information systems</option>
                <option value="4">Software engineering</option>
                <option value="5">Hardware engineering</option></select></td></tr>
            </table>
        </fieldset><br><br><br>
        <fieldset>
<p align='center'><font color=blue size='5pt'><b> QUESTION ON ANY TOPIC COVERED
IN THE E_LEARNING CONTENT: </b></font></p>
            <tablecellspacing="9" cellpadding="2" border="2" width="100%" >
                <tr><td align="center">PLEASE WRITE YOUR QUESTION HERE::<br></td></tr>
                <tr><td align="center"><textarea rows="25" cols="120" name="comments"
></textarea></td></tr>
```

```

        <tr><td align="right">TODAYS DATE:</td><td><input type="text" name="date"
value=<?php echo date("y/m/d"); ?></td></tr>
        <tr><td><input type="submit" name="Formsubmit" value="SubmitForm"
/></td><td><input type="reset" value="ResetForm"></td></tr>
    </table>
    </fieldset><br>
</form>
</body>
</html>

```

Student data processor

```

<?php
$MyName = $_POST['fname'];
$Course = htmlspecialchars($_POST['Course']);
$MyQn = htmlspecialchars($_POST['comments']);
$SubmissionDate = "CURDATE()";

if($_POST['Formsubmit']=="SubmitForm")
{
    $errors = array();
    if(empty($_POST['fname']))
    {
        $errors[] = 'You forgot to enter your name!';
    }
    if(!empty($_POST['col_no']))
    {
        $MyColNo = ($_POST['col_no']);
        $pattern = "/^[A-Z]{1}[\d]{2}\|[\d]{5}\|[\d]{2}$/";
        if (preg_match($pattern, $MyColNo))
        {
            $MyColNo = htmlspecialchars($_POST['col_no']); }

```

```

else{
$errors[] = ' Please enter a valid COLLEGE number!';
    }}
else{
$errors[] = 'You forgot to enter your COLLEGE number!.';
    }
if(!empty($_POST['EMail']))
    {
>Email=($_POST['EMail']);
>pattern = "/^[a-zA-Z0-9._-]+@[a-zA-Z0-9-]+\.[a-zA-Z.]{2,5}$/";
>if (preg_match($pattern, $Email))
    {
>Email= htmlspecialchars($_POST['EMail']);
    }
else{
$errors[] = ' Enter a valid EMAIL!';
    } }
else{
$errors[] = 'You forgot to enter your Email!';
    }
if(empty($_POST['comments'])){
    $errors[] = 'You forgot to enter your QUESTION!';
    }
if (!empty($errors))
    {
>echo '</ul><h3>Your mail could not be sent due to input errors.</h3><hr />';
>// Print each error.
>foreach ($errors as $msg)
    { echo '<li>'. $msg . '</li>';}
?>
<html>

```

```

<body>
<form action="StudentFormFinale.php" method="post" style="padding: 10px; margin: 0px;
background: turquoise;">
<tablecellspacing="10" cellpadding="3" border="2" width="100%" >
<tr><td align="right">Name:</td><td><input type="text" name="fname" value="<?php
echo("$MyName");?>" size="25" maxlength="50"/></td></tr>
<tr><td align="right">Col.No:</td><td><input type="text" name="col_no" value="<?php
echo("$MyColNo");?>" size="15" maxlength="15" /></td></tr>
<tr><td align="right">EMail:</td><td><input type="text" name="EMail" value="<?php
echo("$EMail"); ?>" size="25" maxlength="50"/></td></tr>
<tr><td align="right">Course:</td><td><select name="Course">
    <option value="1" ></option>
    <option value="2" <?php if($Course=="2")echo ("selected=\"selected\");?>>computer
science</option>
    <option value="3" <?php if($Course=="3")echo
("selected=\"selected\");?>>Information systems</option>
    <option value="4" <?php if($Course=="4")echo ("selected=\"selected\");?>>Software
engineering</option>
    <option value="5" <?php if($Course=="5")echo ("selected=\"selected\");?>>Hardware
engineering</option></select></td></tr>
</table>
<tablecellspacing="10" cellpadding="3" border="4" width="100%">
<tr><td align="center">PLEASE WRITE YOUR QUESTION HERE::<br></td></tr>
<tr><td align="center"><textarea rows="25" cols="120" name="comments"><?php
echo("$MyQn");?>" </textarea></td></tr>
<tr><td align="right">TODAYS DATE:</td><td><input type="text" name="date"
value="<?php echo("$SubmissionDate");?>" </td></tr>
<tr><td><input type="submit" name="Formsubmit" value="SubmitForm" /></td><td><input
type="reset" value="Reset form\"></td></tr>
</table>
</form>

```

```

</body>
</html>
    <?php
    }
else if (empty($errorMessage)){
    $db = mysql_connect("localhost", "root","root");
    mysql_select_db("E_LEARNING",$db);
    $sql = "INSERT INTO StudentRecord (RegNo, Contact, Course, Question, SubmissionDate)
    VALUES ('$MyColNo','$Email','$Course','$MyQn', $SubmissionDate)";
    mysql_query($sql,$db) or die(mysql_error());
    echo "<p align='center'><font color=red size='7pt'><b>THANK YOU FOR YOUR
    QUESTION. DO YOU HAVE ANOTHER QUESTION? </b></font></p>";
    ?>
<html><body>
    <form action="StudentFormFinale.php" method="post" style="padding: 10px; margin:
0px; background: turquoise;">
    <fieldset>
    <p align='center'><font color=blue size='7pt'><b> PERSONAL
DETAILS</b></font></p>
    <tablecellspacing="10" cellpadding="3" border="2" width="100%" >
    <tr><td align="right">Name:</td><td><input type="text" name="fname" value=""
size="25" maxlength="50" /></td></tr>
    <tr><td align="right">Col.No:</td><td><input type="text" name="col_no" value=""
size="15" maxlength="15" /></td></tr>
    <tr><td align="right">EMail:</td><td><input type="text" name="EMail" value=""
size="25" maxlength="50"/></td></tr>
    <tr><td align="right">Course:</td><td><select name="Course">
    <option value="1" selected>Choose your Course</option>
    <option value="2">computer science</option>
    <option value="3">Information systems</option>
    <option value="4">Software engineering</option>

```

```

<option value="5">Hardware engineering</option></select></td></tr>
</table>
</fieldset><br><br><br>
<fieldset>
<p align='center'><font color=blue size='7pt'><b> QUESTION ON ANY TOPIC
COVERED IN CLASS </b></font></p>
<tablecellspacing="10" cellpadding="3" border="4" width="100%" >
<tr><td align="center">PLEASE WRITE YOUR QUESTION HERE::<br></td></tr>
<tr><td align="center"><textarea rows="25" cols="120" name="comments"
></textarea></td><tr>
<tr><td align="right">TODAYS DATE:</td><td><input type="text" name="date"
value=<?php echo date("y/m/d"); ?></td></tr>
<tr><td><input type="submit" name="Formsubmit" value="SubmitForm"
/></td><td><input type="reset" value="ResetForm"></td></tr>
</table>
</fieldset><br>
</form>
</body></html>
<?php
    }}
mysql_close($db);
?>
</body>
</html>

```

Lecturers' data

```

<html>
<body>
<form action="COURSEoutline.php" method ="post" style="padding: 10px; margin: 0px;
background: turquoise;">
<fieldset><table cellpadding="3" cellspacing="10" border="4" width="100%" >

```



```

<font size="6" color="blue" align="center"><b> COURSE CONTENT</b></font>
  <tr><td align="right">LECTURER:</td><td><input type="Tinytext" name="Lecturer"
value="" /></td></tr>
<tr><td align="right">Unit Code:</td><td><input type="VARCHAR" name="unitName"
value="" /></td></tr>
<tr><td align="right">Course:</td><td><select name="course">
<option value="1" selected>Choose your Course</option>
<option value="Computer Science">computer science</option>
<option value="Information Systems">Information systems</option>
<option value="Software Engineering">Software engineering</option>
<option value="Hardware Engineering">Hardware engineering</option></select></td></tr>
<tr><td align="right">MAINTOPIC:</td><td><input type="VARCHAR" name="mainTopic"
value="" /></td></tr>
</fieldset><br>
<fieldset>
  <tablecellspacing="10" cellpadding="3" border="4" width="100%" >
    <tr><td ><font size="4" color="red" ><b>
Please enter the subtopics under this topic, INCLUDE KEYWORDS THAT BEST DESCRIBE
THE SUBTOPIC:  </b><br>
YOU <b>MUST </b>separate the Subtopics with<b> a FULL
COLON(:)</b><br></font></td></tr>
<tr><td align="center"><textarea rows="20" cols="80" name="subTopics"
></textarea></td></tr>
<tr><td><input type="submit" name="Formsubmit" value="SubmitForm" /></td><td><input
type="reset" value="ResetForm"></td></tr>
</table>
  </fieldset><br>
</form>
</body></html>

```

Lecturers' data processor

```
<html>
<body>
<?php
if($_POST['Formsubmit']=="SubmitForm")
{
    $errorMessage="";
if(empty($_POST['Lecturer']))
    {
        $errorMessage.="<li>You forgot to enter your NAME!</li>";
    }
if(!empty($_POST['subTopics']))
    {
$STopic= $_POST['subTopics'];
$pattern = "/^[A-Za-z\s]{3,}(:[A-Za-z\s]{3,})+$/";
if (preg_match($pattern, $STopic))
    {
$SubTopics1 = htmlspecialchars($_POST['subTopics']);
    }
else{
$errorMessage.="<li>Please enter SUBTOPICS separated with a : !</li>";
    }}
else
    {
$errorMessage.="<li>You forgot to enter the SUBTOPICS!</li>";
    }
if(!empty($_POST['unitName']))
    {
$unit1= $_POST['unitName'];
$pattern = "/^[A-Za-z]{4}[\s]*[\d]{4}[\s]*$/";
if (preg_match($pattern, $unit1))
```

```

        { $errorMessage="" ; }
else {
$errorMessage.="<li>Please enter a valid UNIT NAME!</li>";
        }}
else {
$errorMessage.="<li>You forgot to enter the UNIT NAME!</li>";
        }
if(empty($_POST['mainTopic']))
        {
        $errorMessage.="<li>You forgot to enter the MAIN-TOPIC name!</li>";
        }
if($course=="1")
        {
        $errorMessage.="<li>You forgot to SELECT the course!</li>";
        }
if(!empty($errorMessage))
        {
        echo("<p><h1>There is an error with the data you have submitted:</h1></p>\n");

        echo("<ul>".$errorMessage."</ul>\n");
?>
<html>
<body>
<form action="COURSEoutline.php" method ="post" style="padding: 10px; margin: 0px;
background: turquoise;">
<fieldset>
<font size="6" color="blue" align="center"><b> COURSE CONTENT</b></font>
        <tablecellspacing="10" cellpadding="3" border="2" width="100%" >

        <tr><td align="right">LECTURER:</td><td><input type="Tinytext" name="Lecturer"
value="<?php echo($_POST['Lecturer']);?>" /></td></tr>

```

```

<tr><td align="right">Unit-Code:</td><td><input type="VARCHAR" name="unitName"
value="<?php echo($_POST['unitName']);?>" /></td></tr>
<tr><td align="right">Course:</td><td><select name="course">
    <option value="1" >Choose your Course</option>
    <option value="Computer Science"<?php if($_POST['course']=="Computer
Science")echo ("selected=\"selected\"");?>>computer science</option>
    <option value="Information Systems"<?php if($_POST['course']=="Information
Systems")echo ("selected=\"selected\"");?>>Information systems</option>
    <option value="Software Engineering"<?php if($_POST['course']=="Software
Engineering")echo ("selected=\"selected\"");?>>Software engineering</option>
    <option value="Hardware Engineering"<?php if($_POST['course']=="Hardware
Engineering")echo ("selected=\"selected\"");?>>Hardware
engineering</option></select></td></tr>
<tr><td align="right">MAINTOPIC:</td><td><input type="VARCHAR" name="mainTopic"
value="<?php echo($_POST['mainTopic']);?>" /></td></tr>
</fieldset><br>
<fieldset>
    <tablecellspacing="10" cellpadding="3" border="4" width="100%" >
        <tr><td ><font size="5" color="red" align="center"><b> PLEASE ENTER THE
SUBTOPICS UNDER THIS TOPIC:          </b><br> YOU <b>MUST </b>separate the
Subtopics with<b> a full colon(:)</b><br></font></td></tr>
<tr><td align="center"><textarea rows="20" cols="80" name="subTopics" ><?php
echo($_POST['subTopics']);?></textarea></td><tr>
<tr><td><input type="submit" name="Formsubmit" value="SubmitForm" /></td><td><input
type="reset" value="ResetForm"></td></tr>
</table>
    </fieldset><br>
</form>
</body></html
<?php
}

```

```

else {
$SubTopics1 = $_POST['subTopics'];
$SUBtopic=explode(':', $SubTopics1);
$num = count($SUBtopic);
$numb = $num-1;
for ($i=0; $i<=$numb; $i++)
    {
// store a single item number and description in local variables
$LectuRER=htmlspecialchars($_POST['Lecturer']);
$unit= htmlspecialchars($_POST['unitName']);
$couRSE=htmlspecialchars($_POST['course']);
$main=htmlspecialchars($_POST['mainTopic']);
$subB = $SUBtopic[$i];
$sub =str_replace("'", "", $subB);
$db = mysql_connect("localhost", "root","root") or die(mysql_error());
mysql_select_db("E_LEARNING",$db) or die(mysql_error());

$query = "INSERT INTO Course_Outline (LECTURER, UnitName, Course,
MainTopic, SubTopic) VALUES ('$LectuRER', '$unit', '$couRSE', '$main', '$sub)";
mysql_query($query ) or die(mysql_error());
    }
echo "<h1>THANK YOU. Your data has been captured, are there more TOPICS? </h1>";
?>
<html>
<body>
<form action="COURSEoutline.php" method ="post" style="padding: 10px; margin: 0px;
background: turquoise;">
<fieldset>
<font size="6" color="blue" align="center"><b> COURSE CONTENT</b></font>
    <tablecellspacing="10" cellpadding="3" border="2" width="100%" >

```

```

        <tr><td align="right">LECTURER:</td><td><input type="Tinytext" name="Lecturer"
value="<?php echo($_POST['Lecturer']);?>" /></td></tr>
<tr><td align="right">Unit-Code:</td><td><input type="VARCHAR" name="unitName"
value="<?php echo($_POST['unitName']);?>" /></td></tr>
<tr><td align="right">Course:</td><td><select name="course">
    <option value="1" >Choose your Course</option>
    <option value="Computer Science"<?php if($_POST['course']=="Computer
Science")echo ("selected=\"selected\"");?>>computer science</option>
    <option value="Information Systems"<?php if($_POST['course']=="Information
Systems")echo ("selected=\"selected\"");?>>Information systems</option>
    <option value="Software Engineering"<?php if($_POST['course']=="Software
Engineering")echo ("selected=\"selected\"");?>>Software engineering</option>
    <option value="Hardware Engineering"<?php if($_POST['course']=="Hardware
Engineering")echo ("selected=\"selected\"");?>>Hardware
engineering</option></select></td></tr>
<tr><td align="right">MAINTOPIC:</td><td><input type="VARCHAR" name="mainTopic"
value="" /></td></tr>
</fieldset><br>
<fieldset>
    <tablecellspacing="10" cellpadding="3" border="4" width="100%" >
        <tr><td ><font size="5" color="red" align="center"><b> PLEASE ENTER THE
SUBTOPICS UNDER THIS TOPIC:          </b><br> YOU <b>MUST </b>separate the
Subtopics with<b> a full colon(:)</b><br></font></td></tr>
<tr><td align="center"><textarea rows="20" cols="80" name="subTopics"
></textarea></td></tr>
<tr><td><input type="submit" name="Formsubmit" value="SubmitForm" /></td><td><input
type="reset" value="ResetForm"></td></tr>
</table>
    </fieldset><br>
</form>
</body></html>

```

```

<?php
    }}
?>
</body>
</html>

```

Problem analyzer

```

<html>
<body>
<?php
require_once "COURSEoutline.php";
require_once "StudentFormFinale.php";
$con = mysql_connect("localhost", "root","root");
if (!$con)
    {
die('Could not connect: ' . mysql_error());
    }
mysql_select_db ("E_LEARNING", $con);
$result = mysql_query("SELECT * FROM StudentRecord") or die(mysql_error());
$numrows = mysql_num_rows($result);
if( $numrows< 1)
    {
echo "<p align='center'><font color=blue size='7pt'><br><br><b>SORRY.<br> THERE ARE
NO QUESTIONS REQUIRING YOUR ATTENTION</b></font></p>";
    }
while($row= mysql_fetch_array($result))
{//start while
$myqn=$row['Question'].":.";
        $myqns=explode(":", $myqn);
        foreach($myqns as $qns)
{//start foreach

```

```

if ($qns!=""){
$query = "SELECT * , MATCH (SubTopic) AGAINST ('$qns' ) AS relevance FROM
Course_Outline WHERE MATCH (SubTopic) AGAINST ('$qns')
ORDER BY RELEVANCE DESC LIMIT 1";
$numresults = mysql_query ($query)or die(mysql_error());
$row = mysql_fetch_array($numresults);
if ($row)
{
$lec = $row['LECTURER'];
$unit = $row['UnitName'];
$ssCourse = $row['Course'];
$M_topic = $row['MainTopic'];
$S_topic = $row['SubTopic'];
$sql = "INSERT INTO ANALYZER ( LECTURERs, UNIT_NAME, COURSE,
MAIN_TOPIC, SUB_TOPIC) VALUES ('$lec', '$unit','$ssCourse','$M_topic', '$S_topic')";
mysql_query($sql) or die(mysql_error());
}
else {
$sql = "INSERT INTO ANALYZER (LECTURERs, UNIT_NAME, COURSE, MAIN_TOPIC,
SUB_TOPIC) VALUES ('Unknown', 'Unknown','Unknown', 'Unknown', 'Not in Syllabus' )";
mysql_query ($sql) or die(mysql_error());
}
} //end if
} //end foreach
} //end while
$row2 = mysql_query("SELECT LECTURERs, UNIT_NAME, COURSE, MAIN_TOPIC,
SUB_TOPIC, COUNT(*) FROM ANALYZER GROUP BY LECTURERs, UNIT_NAME,
COURSE, MAIN_TOPIC, SUB_TOPIC order by COUNT(*) DESC");
echo "<p align='center'><font color=blue size='4pt'><br><br><b>Most of the questions asked
by the students came from the following subtopics.<br> They have been arranged in descending
order.</b></font></p>";

```



```

echo "<table border='1'\>\n";
echo
"<td>LECTURER</td><td>UnitName</td><td>Course</td><td>MAINTOPIC</td><td>SUBT
OPIC</td><td>TOTAL_QNS</td>\n";
echo "<br />";
while($row3 = mysql_fetch_array($row2))
{
    echo "<tr><td>"; echo $row3['LECTURERS'];    echo "</td>";
    echo "<td>"; echo $row3['UNIT_NAME']; echo "</td>";
    echo "<td>"; echo $row3['COURSE'];    echo "</td>";
    echo "<td>"; echo $row3['MAIN_TOPIC']; echo "</td>";
    echo "<td>"; echo $row3['SUB_TOPIC'];    echo "</td>";
    echo "<td>"; echo $row3['COUNT(*)'];    echo "</td></tr>\n";
}
//mysql_query("TRUNCATE TABLE StudentRecord");
mysql_close($con);
?>
</body>
</html>

```

APPENDIX B: Interview Check List

THE QUESTIONNAIRE

1. If you were to be asked questions by a student, what information would you like to know about the student?
2. Have you ever handled a course that deals with electricity and magnetism?
3. How many times have you taught the Unit?
4. When was the last time you handled the unit?
5. In handling the subtopic (state each subtopic in turn from the course content outline in turn) what words do you think are core in describing the content covered?

APPENDIX C: USER MANUAL

Installation

This system requires installation of MySQL and PHP for it to run. These are software that in real life implementation of the system would be installed on the server side and hence users would not require to know about them.

This is followed by installation of the tables, server side pages and input folders required for the system to run

APPENDIX D: STOPWORDS

The following table shows the default list of full-text stopwords (Widenius, Axmark, & AB).

a's	able	about	above	according
accordingly	across	actually	after	afterwards
again	against	ain't	all	allow
allows	almost	alone	along	already
also	although	always	am	among
amongst	an	and	another	any
anybody	anyhow	anyone	anything	anyway
anyways	anywhere	apart	appear	appreciate
appropriate	are	aren't	around	as
aside	ask	asking	associated	at
available	away	awfully	be	became
because	become	becomes	becoming	been
before	beforehand	behind	being	believe
below	beside	besides	best	better
between	beyond	both	brief	but
by	c'mon	c's	came	can
can't	cannot	cant	cause	causes
certain	certainly	changes	clearly	co
com	come	comes	concerning	consequently
consider	considering	contain	containing	contains
corresponding	could	couldn't	course	currently
definitely	described	despite	did	didn't
different	do	does	doesn't	doing
don't	done	down	downwards	during

each	edu	eg	eight	either
else	elsewhere	enough	entirely	especially
et	etc	even	ever	every
everybody	everyone	everything	everywhere	ex
exactly	example	except	far	few
fifth	first	five	followed	following
follows	for	former	formerly	forth
four	from	further	furthermore	get
gets	getting	given	gives	go
goes	going	gone	got	gotten
greetings	had	hadn't	happens	hardly
has	hasn't	have	haven't	having
he	he's	hello	help	hence
her	here	here's	hereafter	hereby
herein	hereupon	hers	herself	hi
him	himself	his	hither	hopefully
how	howbeit	however	i'd	i'll
i'm	i've	ie	if	ignored
immediate	in	inasmuch	inc	indeed
indicate	indicated	indicates	inner	insofar
instead	into	inward	is	isn't
it	it'd	it'll	it's	its
itself	just	keep	keeps	kept
know	known	knows	last	lately
later	latter	latterly	least	less

lest	let	let's	like	liked
likely	little	look	looking	looks
ltd	mainly	many	may	maybe
me	mean	meanwhile	merely	might
more	moreover	most	mostly	much
must	my	myself	name	namely
nd	near	nearly	necessary	need
needs	neither	never	nevertheless	new
next	nine	no	nobody	non
none	noone	nor	normally	not
nothing	novel	now	nowhere	obviously
of	off	often	oh	ok
okay	old	on	once	one
ones	only	onto	or	other
others	otherwise	ought	our	ours
ourselves	out	outside	over	overall
own	particular	particularly	per	perhaps
placed	please	plus	possible	presumably
probably	provides	que	quite	qv
rather	rd	re	really	reasonably
regarding	regardless	regards	relatively	respectively
right	said	same	saw	say
saying	says	second	secondly	see
seeing	seem	seemed	seeming	seems
seen	self	selves	sensible	sent

serious	seriously	seven	several	shall
she	should	shouldn't	since	six
so	some	somebody	somehow	someone
something	sometime	sometimes	somewhat	somewhere
soon	sorry	specified	specify	specifying
still	sub	such	sup	sure
t's	take	taken	tell	tends
th	than	thank	thanks	thanx
that	that's	thats	the	their
theirs	them	themselves	then	thence
there	there's	thereafter	thereby	therefore
therein	theres	thereupon	these	they
they'd	they'll	they're	they've	think
third	this	thorough	thoroughly	those
though	three	through	throughout	thru
thus	to	together	too	took
toward	towards	tried	tries	truly
try	trying	twice	two	un
under	unfortunately	unless	unlikely	until
unto	up	upon	us	use
used	useful	uses	using	usually
value	various	very	via	viz
vs	want	wants	was	wasn't
way	we	we'd	we'll	we're
we've	welcome	well	went	were

weren't	what	what's	whatever	when
whence	whenever	where	where's	whereafter
whereas	whereby	wherein	whereupon	wherever
whether	which	while	whither	who
who's	whoever	whole	whom	whose
why	will	willing	wish	with
within	without	won't	wonder	would
wouldn't	yes	yet	you	you'd
you'll	you're	you've	your	yours
yourself	yourselves	zero		