EDUEASY – SMART LEARNING ASSISTANT SYSTEM

2020-009

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2020

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …………………... Date: …………………….

The supervisor/s should certify the dissertation with the following declaration.

The above candidate has carried out research for the bachelor's degree dissertation under my supervision.

Signature: …………………... Date: …………………….

ABSTRACT

Usage of smart learning concepts has been increased rapidly all over the world recently as better teaching and learning methods. Most of the educational institutes such as universities are experimenting those concepts with their students. Smart learning concepts are especially useful for students to learn better in large classes.

In large classes, lecture method is the most popular method of teaching. In lecture method, the lecturer presents the content mostly using lecture slides and the students make their own notes based on the content presented. However, some students may find difficulties of the above method due to various issues such as speed in delivery.

The purpose of this research is to assist students in large classes in the following content. The research proposes a solution with four components namely note taker, slide matcher, reference finder and question presenter which are helpful for the students to obtain a summarized version of the lecture note, easily navigate to the content and find resources, and revise content using questions.

ACKNOWLEDGEMENT

Working on this project on “EduEasy - Smart Learning Assistant system” was a source of immense knowledge for us. We would like to express my thanks of gratitude to all those who provided me the possibility to complete this project. A special gratitude we give to Dr. Janaka Wijeykoon, the Lecture-in-charge of Comprehensive Design and Analysis Project, Dr. Anuradha Karunasena, our project supervisor & Ms. Pradeepa Bandara, our co-supervisor, whose contribution in stimulating suggestions and encouragement helped us to complete the research project.

The guidance and support received from our families, friends and fellow group members was more helpful to finish the project successfully. Furthermore, we would like to acknowledge all the others who offered their fully support on completion of this research project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
UI	User Interface
API	Application Programming Interface
TF	Term Frequency
SDK	Software Development Kit
SDLC	Software Development Life Cycle
WER	Word Error Rate
HITS	Hyperlink-Induced Topic Search

1. INTRODUCTION

1.1. Background Literature

The smart learning concepts have used rapidly all over the world for better teaching and learning methods [1]. Smart learning reflects how advanced technologies are enabling learners to digest knowledge and skills more efficiently and conveniently. Every day, researchers are seeking out for new technologies and new methods to improve smart learning concepts. So the researchers inclined towards smart learning concepts are growing more than ever before. Although the ongoing global movement towards smart learning makes and important revolution in modern learning, many people still find these concepts vague [2].

Society is normally resistant to change. Since smart learning is in developing stages, it faces several barriers such as rigid nature of traditional education, insufficient time to learn e-learning tools and implement them into teaching practice, lack of recognition, teachers' low self-efficiency in educational technology knowledge, shortage of knowledge and motivation among students [3] etc. However, smart learning can offer many benefits to the academic community. These benefits include increased productivity [4], interactive and enhanced learning experience, simplified teaching, cost reduction etc. The academic community and public at large, should embrace smart learning as solution to many problems.

There are several issues when it comes to a traditional lecture room environment. Sometimes, writing a note in a lecture room environment is difficult because of several reasons [5]. Sometimes the lecturers are speaking too fast. Sometimes students are focused on trying to understand the lecture so they miss to write the note. Since note taking is hard work, transcribing lecturer's voice using a specifically designed program could be explored as a solution.

The component designed for note taking has two main parts. In the first part the component, voice of the lecturer is transcribed to text. There are some previous researches done on this voice transcription part [6]-[7]. In those researches, they use different methods to transcribe audio into text.

The research “Automatic Transcription of Spontaneous Lecture Speech” is a very extensive project [6]. The project was conducted over five years in pursuit of fulfilling three major targets which were building a large scale spontaneous speech corpus, acoustic and linguistic modeling for spontaneous speech recognition and constructing a prototype of a spontaneous speech summarization system. They have trained initial baseline models using a large corpus of lecture presentations & talks and confirmed significant difference of real lectures & written notes. They have proposed sequential decoding and speaking-rate dependent decoding strategies because in spontaneous lecture speech, the speaking rate is generally faster and changes a lot. The sequential decoder simultaneously performs automatic segmentation and decoding of input utterances. Then they have applied the most adequate acoustic analysis, phone models and decoding parameters according to the current speaking rate. Those strategies achieved improvement on automatic transcription of real lecture speech.

“Web-Based Language Modelling for Automatic Lecture Transcription” is a study [7] that propose a Language Model (LM) for lecture transcription which eliminates the need for multiple models, yet achieves similar or better Word Error Rate (WER) reduction. Internet broadcasting is becoming an increasingly popular method of delivering lectures and academic presentations. However, without transcripts, users of online lectures are faced with great difficulties in performing tasks. Automatic Speech Recognition (ASR)

is a cost-effective method to deliver transcriptions, but its accuracy for lectures is not satisfactory. Most lecture recognition systems achieve Word Error Rates (WERs) of about 40-45% [7], quite far from the minimum WER of 25% for a transcript to be useful and accepted by users.

The next part of the component is the text summarization. Automatic text summarization is a complex task which contains many sub-tasks in it. Every subtask has an ability to get good quality summaries. There are also researches done on automatic text summarization, text summarization machine learning algorithms etc. [8]-[10]. They use methods like machine learning algorithms and clustering & extraction to perform text summarizations.

“Graph-Based Algorithms for Text Summarization” is a research [8] that presents innovative unsupervised methods for automatic sentence extraction using graph-based ranking algorithms and shortest path algorithm. Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on information drawn from the graph structure. In this study, they have used two graph-based ranking algorithms which were previously found to be successful on a range of ranking problems. Those algorithms are Hyperlink-Induced Topic Search (HITS) and PageRank algorithms. They can be adapted to undirected or weighted graphs, which are particularly useful in the context of text-based ranking applications. HITS algorithms is a link analysis algorithm that rates web pages. It determines two values for a page, its authority (value of the content of a page) and its hub value (value of its links to other pages). PageRank algorithm is also a link analysis algorithm used by the Google internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents with the purpose of measuring its relative importance within the set. According to this research, Shortest-path algorithm is better because it generates smooth summaries as compared to ranking algorithms.

In comparison, TextRank works well because it does not only rely on the local context of a text unit, but rather it takes into account information recursively drawn from an entire graph. Through the graphs it builds on texts, TextRank identifies connections between various entities in a text and implements the concept of recommendation. An important aspect of TextRank is that it does not require deep linguistic knowledge, domain or language specific annotated corpora, which makes it highly portable to other domains, genres or languages. On the other hand, the Shortest-path algorithm is easy to implement and relatively language independent. Researchers states that the summaries they generated using Shortest-path algorithm are often somewhat smooth to read. When it comes to including the important facts from the original text, the weighting of sentences using traditional extraction weighting methods seems to be the most important part and this Shortest-path algorithm has performed it well in tests. After an extensive comparison, the researchers had come into conclusion that the Shortest-path algorithm is better because it generates smooth summaries as compares to ranking algorithms.

In the research “A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm” [9], a frequent term based text summarization algorithm is designed and implemented in java. This algorithm was implemented using open source technologies like java, DISCO, Porters stemmer etc. and verified over standard text mining corpus. According to the research, this designed algorithms works in three steps. In the first step, the document required to be summarized is processed by eliminating stop words and by applying stemmers. In the second step, term-frequent data is calculated from the document and frequent terms are selected and for these selected words, the semantic equivalent terms are also generated. Finally in the third step, all the sentences in the document, which are containing the frequent and semantic equivalent terms are filtered for summarization.

According to the research paper [9], semantic similarity is a concept whereby a set of documents or terms within term lists are assigned a metric based on the likeness of their meaning/semantic content. Various semantic similarity techniques are available which can be used for measuring sematic similarity between text documents. Semantic similarity methods are classified into four main categories which are Edge Counting Methods, Information Content Methods, Feature Based Methods and Hybrid Methods. In this research, semantic similarity of frequent terms is also used to preserve the meaning of original text document in the summarized document.

In this research, the system is divided into three parts which are an input text document, a summarizer algorithm and a summarized text document as output. The summarizer algorithm is further divided into three parts which are the text processing module, frequent terms generation module and the semantically similar terms and sentence filtering module for summarization. The algorithm takes two input parameters, the input text document and number of frequent terms. As the output, it generates a summarized text document along with the two measures compression ratio, which means how much shorter the summary is than the original and retention ratio, which means how much of the central information is retained.

The research “Extractive Text Summarization Using Sentence Ranking” [10] demonstrates a novel statistical method to perform an extractive text summarization on a single document. In this attempt, sentences are ranked by assigning weights and they are ranked based on their weights. Highly ranked sentences are extracted from the input document so it extracts important sentences which directs to a high-quality summary of the input document. According to this research, there are basically two methods of text summarization. First one is Extractive summarization and the second one is Abstractive summarization. Extractive summarization is basically creating a summary based on strictly what you get in the text. It can be compared to copying down the main points of a text without any modification to those points and rearranging the order of that points and the grammar to make more sense out of the summary. Abstractive summarization techniques tend to mimic the process of ‘paraphrasing’ from a text than just simply summarizing it. Texts summarized using this technique looks more human-like and produce more condensed summaries.

In the proposed approach of this paper, researchers use extractive method to get summary of a given output. Extractive text summarization is divided into two phases as Pre-processing and Processing. They have divided this pre-processing and processing stages into a few steps. Firstly, the file which is given as input is tokenized in order to get tokens of the terms. Then the stop words are removed from the text after tokenization. The words which remained are considered as key words. Then key words are taken as an input and a part of tag is attached to each key word. After completing this pre-processing stage, frequency of each keyword is calculated. As an example, they calculate how frequently a particular key word has occurred and from this, maximum frequency of the keyword is taken. Then, the weighted frequency if the word is calculated by dividing frequency of the key words by maximum frequency of the key words. In this step, the sum of weighted frequencies is calculated. Finally, summarizer extracts the high weighted frequency sentences and the extracted sentences are converted into audio form. The research concludes that the proposed model improves the accuracy when compared to traditional approaches.

One of the researches [31] had explained how to use LCA-based keyword search for effectively retrieving “information unit” from web pages. The authors had discussed the problems of identifying the “Information Unit” of relevant pages containing all the input keywords as the answer. So they take a set of most related web pages as the search target and then model it as a tree structure. For each set of web pages that contain all the given keywords, the smallest and most meaningful subset is selected as a whole to answer the query. Then they come up with an LCA-based (Lowest Common Ancestor) algorithm to identify the most compact subset of the web pages containing all the keywords**.**

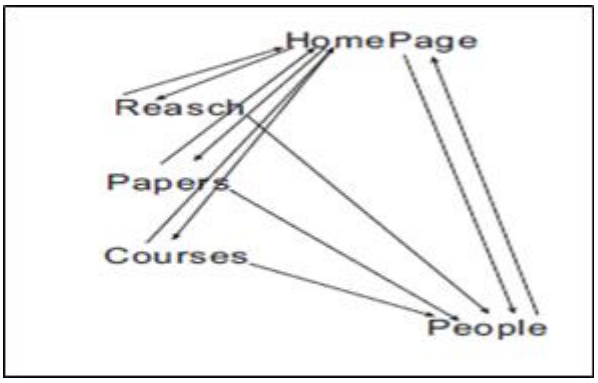


Figure 1.1.1: Link relation inside a small Cluster

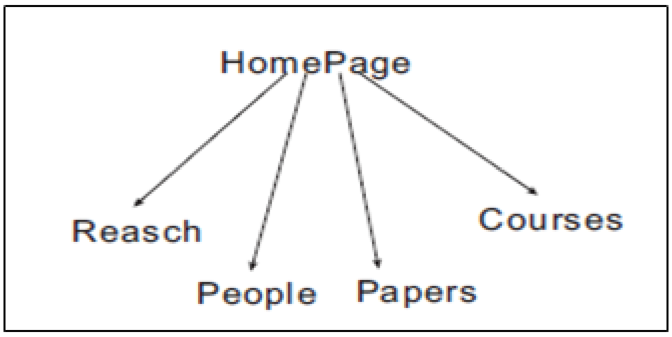


Figure 1.1.2: Breadth-First Traversed Tree

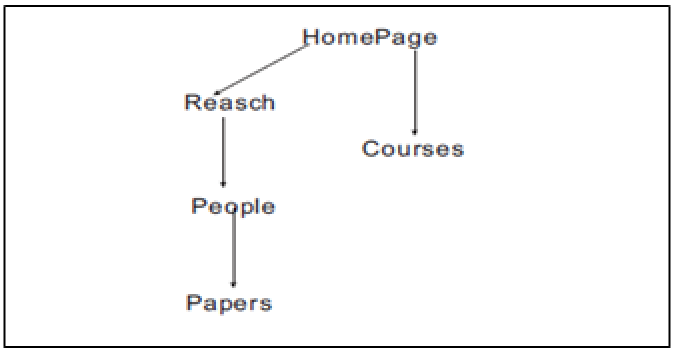


Figure 1.1.3: -Depth First Traversed Tree

There are two ways to traversing the Cluster respectively are Depth First traversal (DFT) and Breadth-First traversal (BFT). Figure 1.1.2 and figure 1.1.3 show the output of the two methods traversing the Cluster shown in figure 1.1.1. Since they store the

Dewey Code of each node in the server, the BFT method will cost less space than the DFT method, as a result, the former method is preferred. Dewey Code is based on Dewey Decimal Classification developed for general knowledge classification. With Dewey Code, each node is assigned a code that represents the path from the root of the tree to the node. Each component of the path represents the local order of an ancestor node. They have used two algorithms for this process. In the first algorithm, they have presented how to model a tree from cluster and encoding with Dewey Code. In the second algorithm, they have presented the LCA-Based algorithm of retrieving "Information Unit" from a Cluster.

The Question & Answer (QA) system finds exact important questions online relevant to different parts of the lectures and whole lecture. Using this application student can get relevant WH questions according to summarized note. Automatic Question Generation Using Natural Language Processing [19] developed a system in which take an input in form of the text file or paragraph from user which contains of the text upon which the user desires to fetch questions; the output is produced in form of a text file or paragraph containing questions based on Bloom's taxonomy. This program takes a text file as an input and generates questions by analyzing each sentence and each paragraph. The importance of generating questions based on Bloom's taxonomy helps students to produce questions that help to test their learning abilities. By deploying agents, the suggested system allows to create queries, agents perform different operations such as paper preparation, detection of information and generation of queries. So the system may also be called a multi-agent system. The tree tagger method and stemming method is used in Paper Preparation to remove the human method. Knowledge classification takes a list of Data Processing produced keywords and seeks the group of such terms in the Bloom by finding the relevant action verb in the repository that matches the given keyword. Question generation module takes the result of classifying knowledge as data to produce queries. The method is a prototype-based approach, which matches the specified keywords according to the degree of the Bloom in the question prototype. A paragraph will be taken as input by the Question Presenter and will produce relevant questions from the key sentences derived from the paragraph. It will be generating different types of WH questions from those selected sentences and paragraphs. The questions will be generated from both simple and complex sentences according to the given text. Significant information about the content of a text is presented by key phrases. Two techniques have been tested for the automated retrieval of key phrases. In order to train the machine, supervised strategies require labeled data which appear to be more precise but also more limited. Unsupervised approaches do not require training sets and tend to be applied to broader areas of expertise, but they are also less essential.

A new method for key phrase extraction called GenEx was also developed by Turney [28], which is based on a series of parameterized heuristic rules tailored to a genetic algorithm. Frank et al. [29] applied a key phrase extraction classifier for Naive Bayes on the same data Turney used which improved the results. Examples of supervised approaches to key phrase extraction include both GenEx and the Naive Bayes classifier. Supervised methods usually require annotated training set which is often not feasible.

Main phrases are sentences (sometimes single words) that are the best goals for question generation (by any measure). Various methods for defining these main words have been introduced. Until delving into query generation at sentence level, one method borrows techniques from automated summarization to define all the primary phrases from an entire document [23]. Another way is to accept main phrases as subject, item, and prepositional phrases comprising a named entity [24]. Other words, such as adverbials, have also been expanded to use the latter approach [25].

Automatic question generation focused on discourse connectives[30], question generation framework divided into two modules collection of content and forming of questions. Content collection consists of identifying the appropriate element of the question from the text to the frame, while question creation includes the disambiguation of the relations of the argument, the recognition of the sort of question and the introduction of syntactic changes to the material. Researcher focuses on seven connective discourses like since, as a result , for example, and on that basis, question form can be determined as if sentence consists because then question type will be "Why." The method was tested by two assessors for the semantic and syntactic soundness of the question.

For self-directed learning, the textual query generator is of critical importance. It has been demonstrated [26] that interviewing is an efficient way to make learners understand more. Unfortunately, multiple studies have shown that students are mostly oblivious of their own expertise, barely ask questions and prefer to ask superficial questions [27]. From a broad range of channels, learners have access to learning materials, and these materials are not always accompanied by questions that help facilitate learning.

With no standardization, current tests of relational keyword search frameworks are ad hoc. Webber [22] describes current search-effectiveness tests. While the criterion was created by Coffman and Weaver [20], their analysis did not provide any success appraisal. A. Baid [16] believes that many current keyword search strategies are behaving unpredictably due to unacceptable1response times or struggling to generate results even after exhausting memory. This argument is supported by the results, especially by the broad memory footprint of the systems. A number of hierarchical keyword search frameworks beyond those used in this review have been released. L. Chang [19] and Kshirsagar and Sudarshan [18] all offered keyword search tutorials inside databases. X. Yu offers an outstanding description of the methods used in reference keyword search.

One problem which has become very common among researchers is the problem of creating a completely functioning question answering method. An implementation of a QA presenter usually scans Google and shows a correct short answer. Most of the researchers working in different domains such as Data Mining, NLP, Knowledge Recovery and Information Extraction have been concentrating on open-domain QA and close-domain QA method for application of the QA method. Based on the topic, programs can be classified into two types; Closed-domain addressing questions applies to unique topic relevant questions. Open-domain question-answering answers the questions associated with each domain. The work in [17] gives each of the sentences a score to obtain from the text all of the important sentences. Now, to solve the issue, they do similarity matching between the obtained sentences and user queries. This proves to be a reasonably good strategy, but in order to solve the question, the solutions are not necessarily framed properly. The AQUA system [21] divides the sentence into subject, verb, propositional sentences, adjectives and goals among its different steps for the processing of a question. Similar to QACID, it generates a semantic query representation that is used by search algorithms when attempting to locate a reaction in the knowledge base.

Slide Matcher component matches the relevant lecture slide content according to the generated summarized note. It means If student wants to refer the lecture note which is published by the lecturer, can navigate to the lecture note directly using slide matcher tool. Student has to click on the topic or the topic of summarized note and system will display the exact place where the lecture content is.

Finding the exact lecture note according to the summarized note is basically about matching the words and find the similarity of documents.

According to the Cambridge dictionary, Similarity is the fact that people or things look or are the same. Similarity is the state of being almost the same, or a particular way in which something is almost the same. Finding the text similarity between two documents can be done in many ways.

- Using string matching algorithms

String matching algorithms such as Rabin-Karp algorithm, Boyer – Moore String-search algorithms basically matches the strings of a word with each words of other sentence and find the similarity.

- Using Natural Language Processing (NLP) functions

There are some functions in NLP which can be re-processed the data. Re-processing data is very essential to find the similarity of two text documents which are converted to text from voice. Basically Edu-Easy application works with lecturer's voice which is converted to text. Then the converted words can be changed to it's based form or root form and changed prefixes and suffixes. NLP functions such as tokenization, Stemming, Lemmatization will help to overcome those challenges.

- Using tf_idf Vectorization

tf_idf is an abbreviation for Term Frequency – Inverse Document Frequency which is very common algorithm to transform text into numbers and identify the similarity. Tf-idf is a measure of originality of a word by comparing the number of times a word appears in a document with the number of documents the word appears in.

- Using Gensim

Genism is a popular open source natural language processing library which can perform with complex texts.

Matching the relevant lecture slide content according to the generated summarized note is the main objective of this Slide Matcher component. Simply system compares the words of topic which is in the summarized note with lecture content. By developing an algorithm to match the string values can solve the problem. But there can be many failures by only using string matching algorithms. In String matching algorithms, it takes the values into an array and compares with the other variables. But comparing exact words cannot find the best solution. Because EduEasy system converts the lecture's voice into text document, removes the unwanted phrases and summarizes the note. Therefore the topics and subtopics in summarized note might not be exactly same in lecture note.

Then a problem occurs how to compare each words without having same words. There can be the same meaning in different words.   By referring research papers, a solution was found called Natural Language Processing which was found by Richard Bundler and John Grinder. A research [16] done about Natural Language Processing algorithms described deeply about the functions of NLP and how it can be used to re-process the data. NLP is a part of computer science and AI which deals with human languages. Chatbox, Spell Checking , Keyword searching are some applications which used NLP algorithms.

There are many functions in NLP which helps to re-process the data. The research [26] has explained about the main steps and clear solutions to overcome the problems if only string matching algorithms are used to compare the similarity of two sentences. Data cleaning is a function about to clean the data. Get the data in clean and standard format for further analysis is very important. This function can remove the unwanted spaces, commas and etc. After getting the data cleaned, system cannot match the whole document at once. So there is an another function called Tokenization to reduce the complex of sentences. Tokenization can break a complex sentence into words, understand the importance of each word to the sentence and produce a structured description on an input sentence. Next function is Stemming. Stemming does normalize words into its base form or root form, cuts the prefixes and suffixes. Changing affected, affective to affect is an example. There is an another function called Lemmatization which groups together different infected forms of a word called Lemmas and outputs a proper word. Using Lemmatization the words 'go', 'going' and 'gone' will turn into 'go'. Chuncking is an another function which is picking up individual pieces of information and grouping them into bigger pieces such as nouns, verbs and etc. POS tags and Named Entity Recognition are some other functions in NLP which helps to reprocess the data.

An another research [27], published by Aselsan research center in Turkey described about the types of text similarity and how it helps to find the most similar sentences . It describes about mainly two types in text similarity. Lexical similarity and Semantic similarity are the main two types. Lexical similarity is about surface closeness and semantic similarity is about the meaning of the sentence. When we measure the similarity, both similarities should be noticed. An article [28] explained about this problem with an example. There are two sentences called "The cat ate the mouse" and "Mouse ate cat's food". If the word level similarity is only concerned, these two phrases appear very similar as 3 of the 4 unique words same. But when we consider the meaning, it is totally different. To overcome this problem there is an algorithm called Jaccard similarity which is explained in an article [28].

Sentence 1: AI is our friend and it has been friendly
Sentence 2: AI and humans have always been friendly

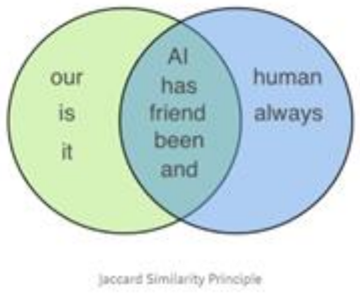
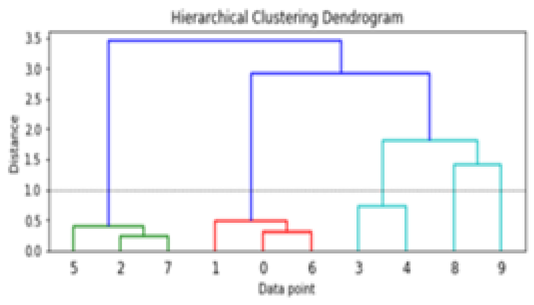
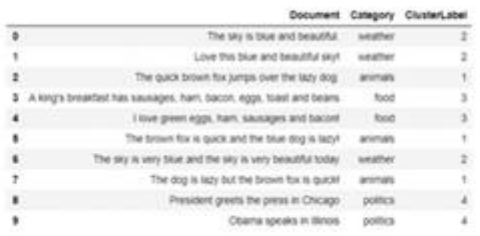


figure 1.1 – Jaccard similarity principle

This figure 1.1 is shown an explanation about how Jaccard similarity works.

There is an another problem may occur when comparing two sentences. There can be two sentences, both sentences have the same meaning but the words used in two sentences are exactly different. Then when we compare the similarity, both sentences do not match with each other. For exam a research paper [29] explained it with an example. “President greets the press in Chicago” and “Obama speaks in Illinois” are those two sentences and to identify the comparison of those two sentences that research paper [29] explains an algorithm called K-mean algorithms and Hierarchical Clustering Dendrogram. Here these two sentences should be converted into vectors first and there are methods explained “Count Vectorizer method” and “Word Embeddings”.





	Document	Category	ClusterLabel
0	The sky is blue and beautiful.	weather	2
1	Love this blue and beautiful sky!	weather	2
2	The quick brown fox jumps over the lazy dog.	animals	1
3	A king's breakfast has sausages, ham, bacon, eggs, toast and beans	food	3
4	I love green eggs, ham, sausages and bacon!	food	3
5	The brown fox is quick and the blue dog is lazy!	animals	1
6	The sky is very blue and the sky is very beautiful today	weather	2
7	The dog is lazy but the brown fox is quick!	animals	1
8	President greets the press in Chicago	politics	4
9	Obama speaks in Illinois	politics	4

Figure 1.2 – Hierarchical clustering dendrogram

Figure 1.3 – Data clustering example

This figure 1.2 is shown about Hierarchical Clustering Dendrogram an figure 1.3 is shown how clustered words which has the same meaning with difference words as “Obama” and “President” are grouped in politics category.

A research paper [28] describes a mathematical algorithm called Cosine vector similarity to calculate the semantic similarity. It is an algorithm which calculates the similarity of a sentence by measuring the cosine angle between two vectors.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 1.4 - cosine vector algorithm

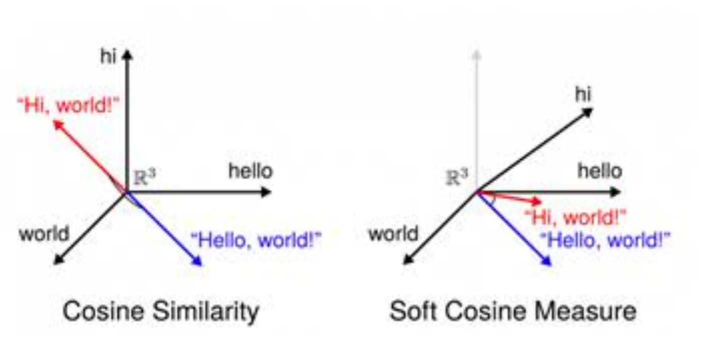


Figure 1.5 – example of cosine vector similarity

This figure 1.5 is explained about Cosine vector similarity and how it can be graphically drawn.

A research paper [16] which was written about text similarity in vector space model, has been described two ways how text similarity can be calculated. Structure based and structure agnostic are the two ways. Structure based means rely on a logical structure of the text and transform it into an intermediate structure. Then the comparison will be done. Second way is Structure based which ignores the structure. A vector Space Model(SVM) is used to represent the text. Then that text is converted into a numeric vector. Research paper[16] described about TFIDF vectorization model which measures the term frequency of each term in a text and multiplies it by the logged inverse document frequency of that term across the entire corpus.

After converting text to a vector, all texts are represented as low dimentiol vectors. Given two vectors can measure similarity in many ways. According to the paper [16] Euclidean distance, angular distance, correlation are some of them. Even though these difference methods measure the similarity in difference ways, research paper [16] adopted Cosine Vector similarity as the well-known and frequently used algorithm. Author discussed about how those algorithms represent to similarity detection power for texts. Doc2Vec variants and Word2Vec variants are compared and author found that Doc2Vec is more superior than Word2Vec. There are less researches conducted about the performances of similarity comparison for longer texts according to the author [16]. But Using TFIDF methods to detect the similarity is quite advanced because it has the advanced word embedding techniques.

1.2. Research Gap

According to the available research papers and resources, there are several systems developed related to smart learning. So far, there is no simple, cohesive concept of smart learning. The definition if smart learning is constantly discussed by multidisciplinary scholars and educational professionals. In general, the atmosphere for intellige3nt learning is productive, efficient and engaging. In this section, the research gap between previous researches/products and the Note Taker component of the EduEasy Smart Learning Assistant System is discussed.

The Note Taker is one of the four main components of the EduEasy Smart Learning Assistant System. This component is designed to do two particular tasks. As the first task, the program converts the lecturers voice in to text and create a voice transcript. Then the second task is creating a summarized note. The program takes the previously created voice transcript and creates a summarized note.

As stated in the previous section, there are many researches done on voice transcriptions [6]-[7] and text summarization [8]-[10]. In those speech and lecture transcription researches, they have used different and novel methods to transcribe audio into text. Also, in those text summarization researches, they have used different type of methods and algorithms to do summarizations but a comparison of effectiveness between those methods is a significant gap. Some researchers have done some calculations to determines the accuracy of the effectiveness of their methods but despite the numbers, a summary should be effective to a person reading that. So the opinion of readers matters the most but researching that area has not been done before.

There are online word processing tools (i.e. LinguaKit) to summarize texts and do speech-to-text transcriptions. But there is no research or product which is a combination of voice transcription and text summarization. So in this Note Taker component of the EduEasy Smart Learning Assistant System, those two features are combined and a full functioning speech-to-text converter and a text summarizer is implemented.

There are some tools and software that are made to find some reference materials. They have different functionalities compare to the EduEasy Reference Finder. The SliePlayer application only provides presentation slides as the output when searching for reference materials. On the other hand, the CSUN Ovit Library only provides e-books and pdf. Both of the applications can not search for references from anywhere on the internet. But both applications show user target recommendations. They have only a limited database for finding references. And also both of the applications do not present every type of reference material and they can not generate the similarity value by comparing two documents as the Reference Finder does.

The Reference Finder can find any type of reference material automatically and it can be searched anywhere on the internet. And also it shows user target recommendations in separate tabs in the search engine. As the most important point, the Reference Finder can generate a similarity value. The above facts show that there is no such application that has all the functionalities similar to the Reference Finder application as one product.

2. RESEARCH PROBLEM

Students who study smart instead of study hard, can have good success in their education. But studying smart is a challenge without a correct support. University students get used to study their lessons in various ways. EduEasy is a smart learning assistant system for students to effectively learn and revise lectures done at the university. The Note Taker is one of the main four components of the EduEasy Smart Learning Assistant System which is designed to serve a specific purpose.

Most of the time university students do not have a proper lecture note for various reasons. Some students do not attend lectures regularly or even if they do attend lectures, they may not write down important points etc. Students who do not have proper notes may feel uncomfortable during exam periods because without a good note, it is very difficult to understand theories and other content. Then they have to waste their time finding notes instead of studying. The Note Taker component is designed to provide a solution for this problem. This application generates a summarized note using a transcription of the lecturer's voice. This speech-to-text converting facility is also provided by the application. After that, students are able to revise the notes anytime they want.

Referring the relevant reference materials is very important to all the students when they aim for the best grade in any subject. It helps to understand difficult lessons, formulas, theories, etc. The lecturer can't teach everything regarding a lesson within the two-hour lecture time. So, students should search for more information relating to lessons by themselves.

Most of the students skip the process of understanding lessons by referring to references and they memorize the difficult lessons as it is due to the limited time that they get during exam periods. Sometimes it is hard to find exact relevant reference materials according to the lecture note in a short period. Some students prefer to watch videos to understand difficult lessons rather than reading articles or pdf. Because of that it more important to search for different kinds of reference materials such as YouTube videos, pdf, articles, papers, and books in one go as well. Referring to the reading materials such as articles, pdf, and papers is time-consuming, and as well as it is difficult to determine how much similar the reference material to the lecture note. So there should be a way to choose the most appropriate article to make the time more meaningful than wasting.

3. RESEARCH OBJECTIVES

3.1. Main Objective

- Develop an E-leaning application for students to effectively learn and revise lectures done at the university.

3.2. Specific Objectives

- Capture the lecturer’s voice using microphone and voice recorder.
- Develop a speech-to-text converter to transcribe lectures into text.
- Develop a text summarizer to summarize the transcription.
- Conduct a research/survey to find out the most effective text summarization algorithm among popular algorithms.
- Develop an attractive and easy to use user interface for the speech-to-text converter and the text summarizer.
- Deploy the web application.

4. METHODOLOGY

4.1. Methodology

Students who study smart instead of study hard, can have good success in their education. But studying smartly is a challenge without the correct support. EduEasy is an E-learning application for students to effectively learn and revise lectures done at university. EduEasy application has mainly four components. They are note taker, reference finder, question presenter and slide matcher. Each of those components will help to overcome problems which university students has to face on a daily basis.

University students get used to study their lessons in various ways. Students who do not go lectures regularly may not have good lecture notes. Students who don't have proper notes, may feel uncomfortable during exam periods because without having a good note it is very difficult to understand the theories and learn them. Then they waste their time to find out good notes. EduEasy application is designed to automatically generate a summarized short note by first, recording the lecturer's voice & converting it into a text document and then, using a text summarization algorithm, removing unnecessary information & converting the text document into a summarized note.

The Note Taker component of the EduEasy application is divided into two basic parts. First one is Speech-to-Text converter and the second one is the Text Summarizer. Speech-to-Text converter is designed to convert lecturers voice to text. This process can be explained using a few steps.

First, the lecturer's voice should input through the microphone. This can be done by either real time or recording the voice to another device and play it to the microphone later. Reason for using this method instead of converting an entire lecture recording audio file at once is that, converting large audio file takes a long time than converting real time. So to increase the efficiency of the application, this method was used.

Then the audio is converted to text using JavaScript Web Speech API. The Web Speech API incorporates voice data into web applications. The Web Speech API has two main parts, SpeechRecognition (Speech-to-Text) and SpeechSynthesis (Text-to-Speech). To develop the Speech-to-Text converter function in the Note Taker component, speech recognition is used. Speech recognition is accessed vis the `SpeechRecognition` interface, which provides the ability to recognize voice context from an audio input (normally via the device's default speech recognition service) and respond appropriately. Generally, the interface's constructor is used to create a new `SpeechRecognition` object, which has a number of event handlers available for detecting when speech is input through the device's microphone.

Figure 2.1.1 demonstrates an overview of the Reference Finder component The automatic reference finding process begins after the student select a summarized note which is stored in the system database. Then the Reference Finder extracts the keyphrases from the summarized note and automatically searches the references according to the keyphrases. The references appear in the separate tabs in the search engine and the results contain YouTube videos, web articles, e-books, pdf, etc. Then using one of the reference materials that appear in the search engine, the system compares the content similarity between the summarized note and that web reference. The choosing reference should not be any kind of video material. After that, the similarity algorithm generates a similarity value and the student can get an idea the web results are how much similar to the summarized note. Finally, the student can refer to the most relevant and rated references using the reference finder component.

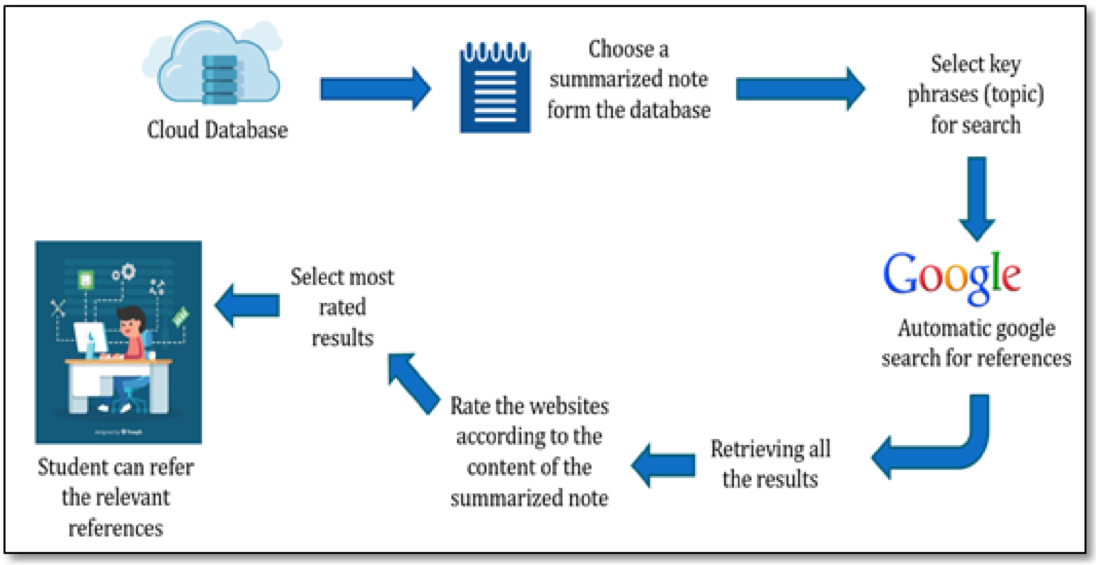
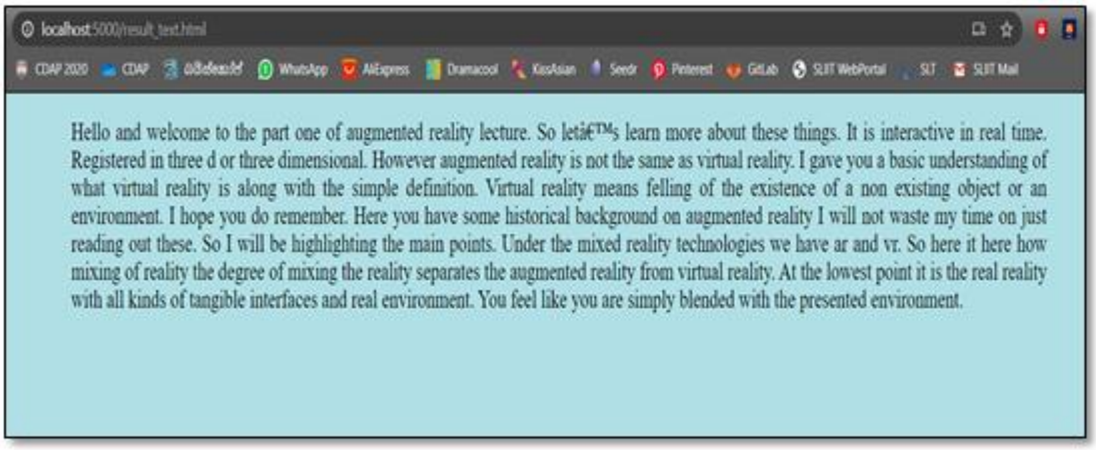

Cloud Database
Choose a summarized note form the database
Select key phrases (topic) for search
Google
Automatic google search for references
Retrieving all the results
Rate the websites according to the content of the summarized note
Select most rated results
Student can refer the relevant references

Figure 2.1.1: The system flow diagram of Reference Finder component

Figure 2.1.2: Tools and technologies

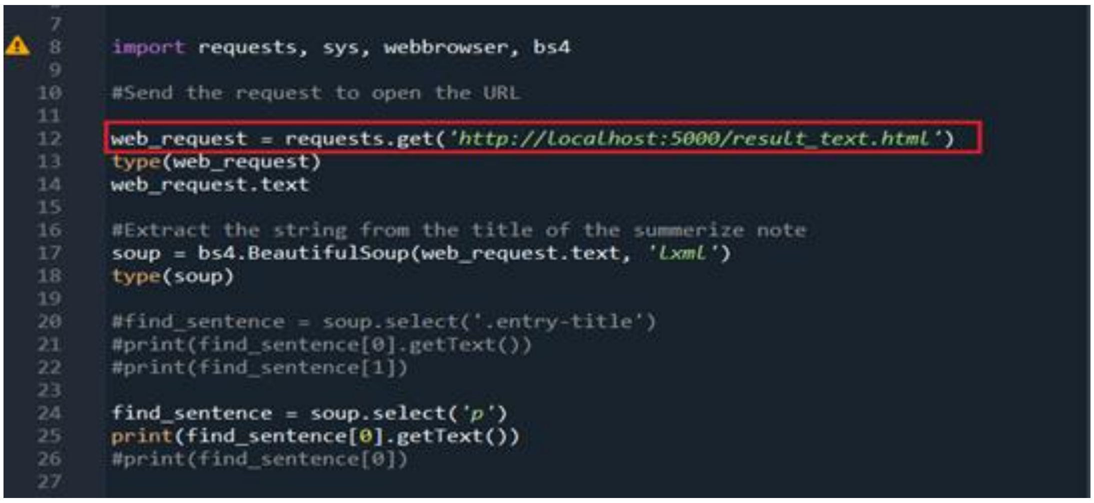
Figure 1.2.2 depicts the tools and technologies used to develop the Reference Finder component. Python version 3.7 is used to develop the whole Automatic Reference Finder application. All the python implementation is done by using “Anaconda Spyder” IDE. Spyder (Scientific Python Development Environment) is a free integrated development environment (IDE) that is included with Anaconda. And also It includes editing, interactive testing, debugging, and introspection features. When developing the web application “JetBrains PhpStorm 2017.2” IDE is used along with HTML, CSS, and Bootstrap. “Flask” is a lightweight WSGI web application framework and it is used to connect the web application with the back-end python functions. “GitLab” is used to manage the EduEasy project and its source files, as they are changing over time.



Hello and welcome to the part one of augmented reality lecture. So letâ€™s learn more about these things. It is interactive in real time. Registered in three d or three dimensional. However augmented reality is not the same as virtual reality. I gave you a basic understanding of what virtual reality is along with the simple definition. Virtual reality means felling of the existence of a non existing object or an environment. I hope you do remember. Here you have some historical background on augmented reality I will not waste my time on just reading out these. So I will be highlighting the main points. Under the mixed reality technologies we have ar and vr. So here it here how mixing of reality the degree of mixing the reality separates the augmented reality from virtual reality. At the lowest point it is the real reality with all kinds of tangible interfaces and real environment. You feel like you are simply blended with the presented environment.

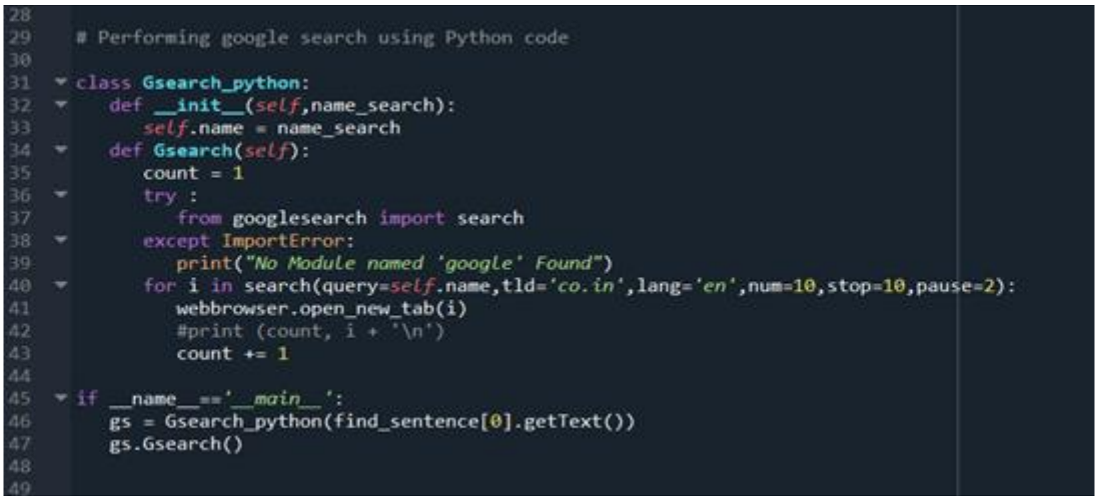
Figure 2.1.3: The input of the Reference Finder

As shown in Figure 2.1.3, this is the input of the Automatic Reference Finder that is generated from the Note Taker component. The web scraping technique is used for the keyphrase extraction process and the automatic google search process.



```
import requests, sys, webbrowser, bs4

#Send the request to open the URL

web_request = requests.get('http://localhost:5000/result_text.html')
type(web_request)
web_request.text

#Extract the string from the title of the summerize note
soup = bs4.BeautifulSoup(web_request.text, 'lxml')
type(soup)

#find_sentence = soup.select('.entry-title')
#print(find_sentence[0].getText())
#print(find_sentence[1])

find_sentence = soup.select('p')
print(find_sentence[0].getText())
#print(find_sentence[0])
```

Figure 2.1.4: Request the summarized note URL to extract the keyphrases



```
# Performing google search using Python code

class Gsearch_python:
   def __init__(self,name_search):
      self.name = name_search
   def Gsearch(self):
      count = 1
      try :
         from googlesearch import search
      except ImportError:
         print("No Module named 'google' Found")
      for i in search(query=self.name,tld='co.in',lang='en',num=10,stop=10,pause=2):
         webbrowser.open_new_tab(i)
         #print (count, i + '\n')
         count += 1

if __name__=='__main__':
   gs = Gsearch_python(find_sentence[0].getText())
   gs.Gsearch()
```

Figure 2.1.5: Automatic Reference Finder

According to Figure 2.1.4 first, import the requests library, sys library, bs4 library, and webbrowser library to the project. Then copy the URL of the summarized note and paste it in front of the “web_request” method. The requests library is used for requesting the URL. After that using bs4 (BeautifulSoup library) library the application can extract any content that is inside the particular HTML tags or CSS classes. The contents are extracted in the paragraph tags of the summarized note as shown in Figure 1.2.4. The extracted content is saved under the “find_sentence[0].getText()” variable. According to Figure 1.2.5, using the “find_sentence[0].getText()” as the input for the automatic reference finder application, it searches the references and opens them in separate tabs in the search engine. The webbroweser library is used to open the tabs in the search engine.

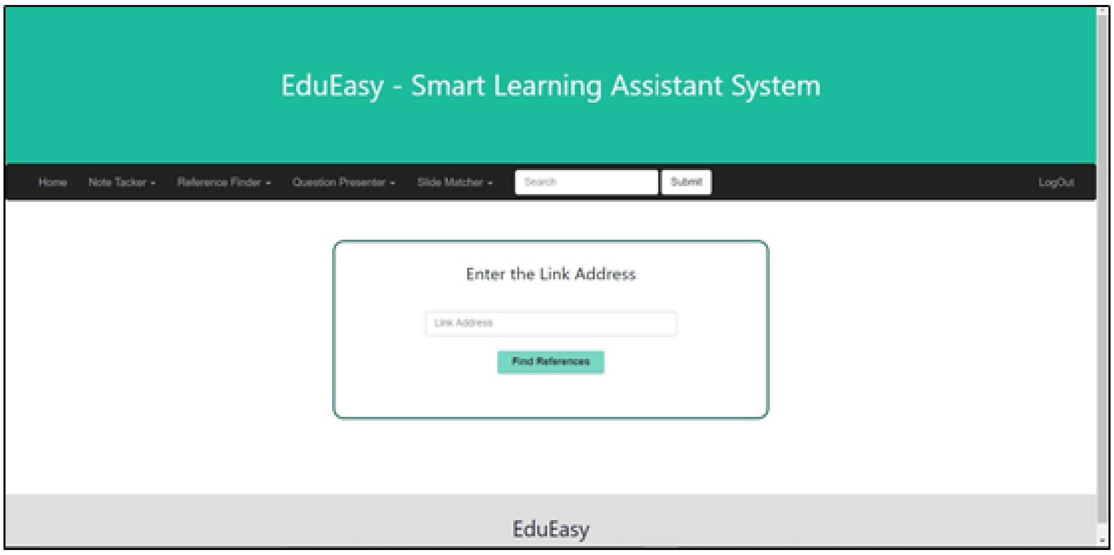


Figure 2.1.6: User Interface of "Finding References"

Figure 2.1.6 depicts the user interface of the first part of the Reference Finder component.

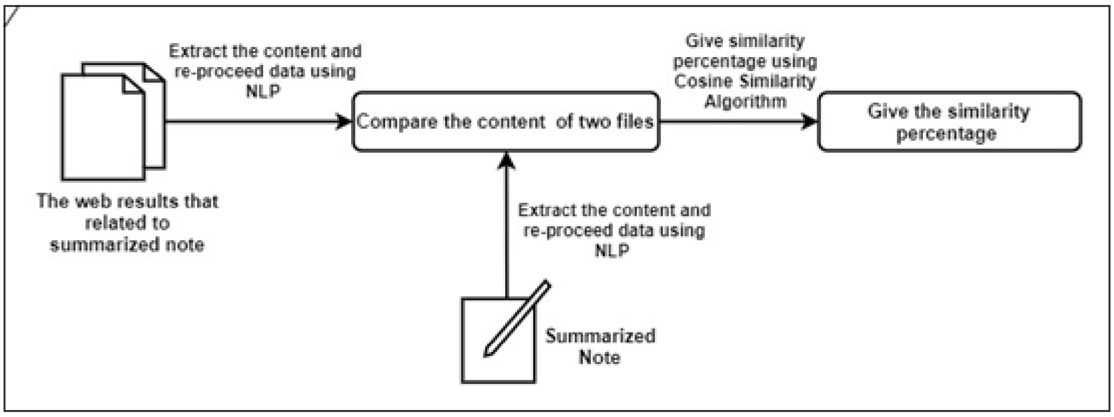
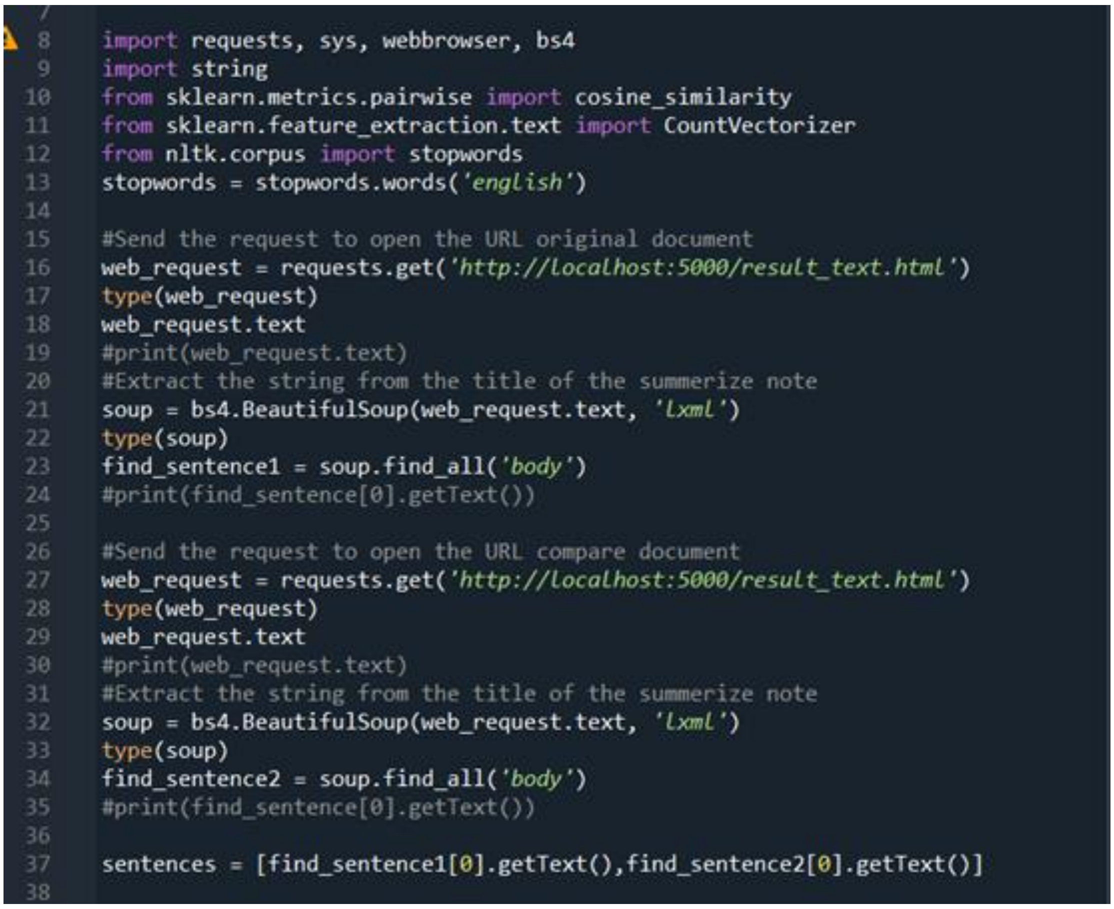

Extract the content and re-proceed data using NLP
Give similarity percentage using Cosine Similarity Algorithm
Compare the content of two files
Give the similarity percentage
The web results that related to summarized note
Extract the content and re-proceed data using NLP
Summarized Note

Figure 2.1.7: System diagram for the second part of the Reference Finder component

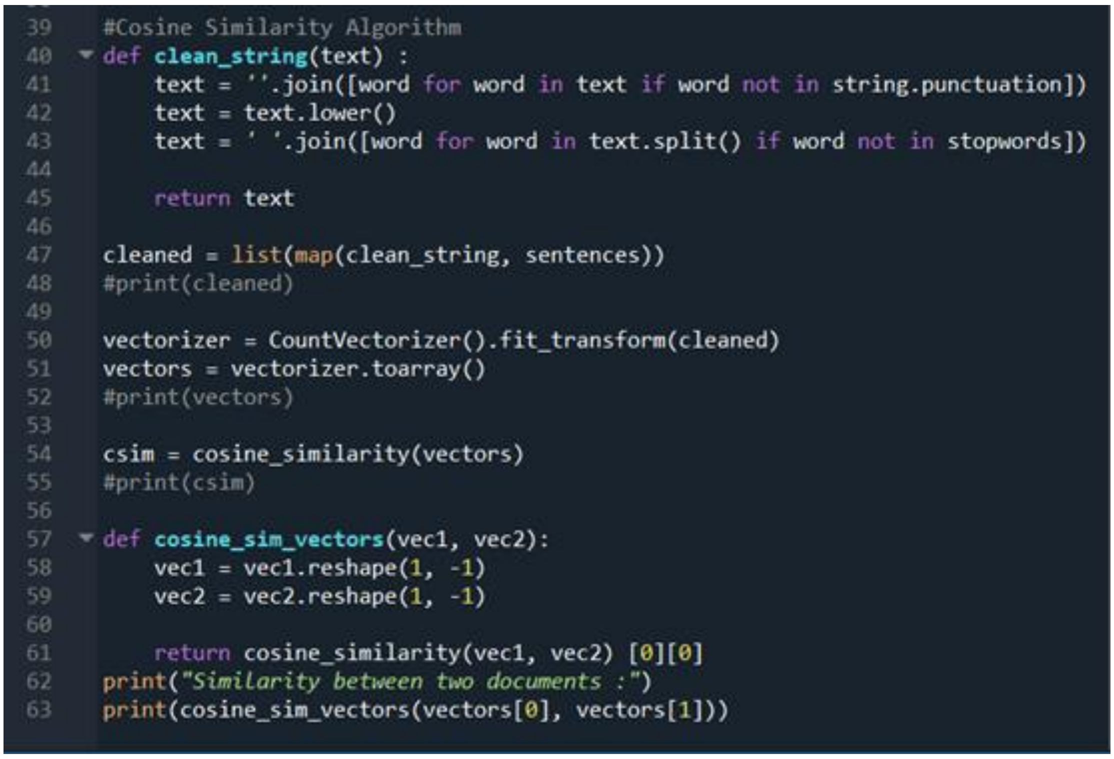
Figure 2.1.7 demonstrates how the second part of the Reference Finder Component is worked. The second half of the Reference Finder component begins with inputting the summarized note's URL and the one of the reference material's URL into the application as shown in Figure 2.1.8. The WebScrapping technique is used to do this also. Then the application extracts the whole content of the two web pages and stores them separately under two variables.



```
import requests, sys, webbrowser, bs4
import string
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
stopwords = stopwords.words('english')

#Send the request to open the URL original document
web_request = requests.get('http://localhost:5000/result_text.html')
type(web_request)
web_request.text
#print(web_request.text)
#Extract the string from the title of the summerize note
soup = bs4.BeautifulSoup(web_request.text, 'lxml')
type(soup)
find_sentence1 = soup.find_all('body')
#print(find_sentence[0].getText())

#Send the request to open the URL compare document
web_request = requests.get('http://localhost:5000/result_text.html')
type(web_request)
web_request.text
#print(web_request.text)
#Extract the string from the title of the summerize note
soup = bs4.BeautifulSoup(web_request.text, 'lxml')
type(soup)
find_sentence2 = soup.find_all('body')
#print(find_sentence[0].getText())

sentences = [find_sentence1[0].getText(),find_sentence2[0].getText()]
```

Figure 2.1.8: Get the two URLs to compare the similarity

Before going through the compare content similarity process the content data have to re-process using Tokenization and removing stop words techniques. Finally, according to Figure 2.1.9, the Reference Finder compares the content of the summarized note and the web results that are generated as the reference in google and generates a similarity value using the cosine similarity algorithm.



```python
#Cosine Similarity Algorithm
def clean_string(text) :
    text = ''.join([word for word in text if word not in string.punctuation])
    text = text.lower()
    text = ' '.join([word for word in text.split() if word not in stopwords])

    return text

cleaned = list(map(clean_string, sentences))
#print(cleaned)

vectorizer = CountVectorizer().fit_transform(cleaned)
vectors = vectorizer.toarray()
#print(vectors)

csim = cosine_similarity(vectors)
#print(csim)

def cosine_sim_vectors(vec1, vec2):
    vec1 = vec1.reshape(1, -1)
    vec2 = vec2.reshape(1, -1)

    return cosine_similarity(vec1, vec2) [0][0]
print("Similarity between two documents :")
print(cosine_sim_vectors(vectors[0], vectors[1]))
```

Figure 2.1.9: Compare similarity algorithm

Here vectors are used to calculate the similarity value. If the user inputs the same URL as the two inputs the similarity value must be 1.000. If the user inputs different URLs that are relevant to different topics (as an example one document can be related to data mining and the other document can be related to heart patients) the similarity value must be 0.000. Finally, the student can refer to the most relevant and rated references using the reference finder component.

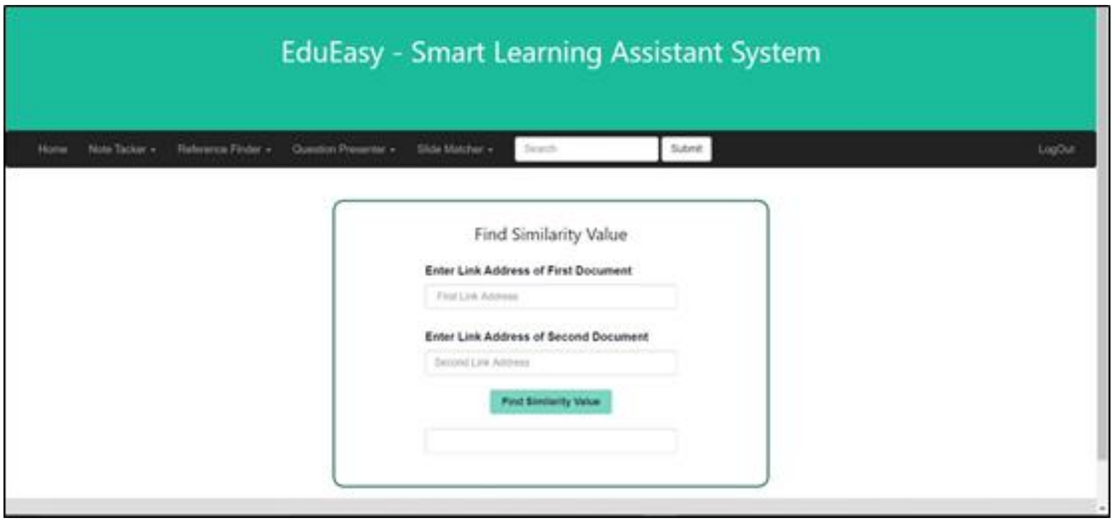


Figure 2.1.10: User Interface of “Similarity Value”

Figure 2.1.10 depicts the user interface of the second part of the Reference Finder component.

In this implemented system, "Question Presenter," we will use Hierarchical Keyword Search technique to render Sophisticated Information Retrieval System and produce questions using paragraphs. Current system in which many existing search techniques for practical retrieval tasks do not produce sufficient results. Memory management consists of several retrieval methods, in particular, programs. In previous assessments, we will illustrate the relationship between execution time and various variables; our analysis suggests that these variables have moderately little output dispute. In short, our study would affirm the previous argument that these strategies have unacceptable operating efficiency and point out the need for accuracy as represented by the NL region when we are going to analyze these recovery systems.

Students may want to send any questions and answers about the related section when referring to the summarized note. In search engines, students should then have to find sample questions by manually searching. Most students are unable to find the best answers to these questions. The student's time would then waste, and they will not find sample questions and responses at times. The EduEasy application has a method of specifically searching the questions and answers. Sample questions about the related lecture element are explicitly given by the Question & Answer presenter section. What's crucial is that the QA presenter gives the right short answers to the given questions.

“Question Presenter” component can be divided two main objectives. The main objectives of this component will be to Generate Questions based on the summarized note (Automatic question generation by using NLP). The next main objective of this component is to search online answers for relatable questions using keywords.

This component 's main objective is to produce questions based on the summarized notice (Automatic generation of questions using NLP). A text file forwarded to the program as argument. By interpreting each sentence, this program takes a summary text file as an input and produces questions. This summarized note or paragraph file is read using a python package called text bolt. And also each paragraph is further broken down into sentences using the function **`parse(string).`** `And also, each sentence is passed as string to function` **`genQustion (line).`**

These are the part-of-speech tags which is used in this system.

NNS	Noun, plural
JJ	Adjective
NNP	Proper noun, singular
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBZ	Verb, 3rd person singular present
VBD	Verb, past tense
IN	Preposition or subordinating conjunction
PRP	Personal pronoun
NN	Noun, singular or mass

With the use of condition statements, each sentence is parsed using English grammar rules. A dictionary named bucket is developed, and part-of - speech tags are attached to it. A query sentence is successfully produced by the sentence that gets parsed. The produced list of questions is printed as production. The input text and the query produced below are given.

-----------INPUT TEXT-------------

Flute is an Indian classical instrument. Akhil plays Flute and Guitar.

Polsambol is a Sri Lankan dish made of rice and tamarind.

Mahagamasekara writes books.

Osmosis is the movement of a solvent across a semipermeable membrane toward a higher concentration of solute. In biological systems, the solvent is typically water, but osmosis can occur in other liquids, supercritical liquids, and even gases.

When a cell is submerged in water, the water molecules pass through the cell membrane from an area of low solute concentration to high solute concentration. For example, if the cell is submerged in saltwater, water molecules move out of the cell. If a cell is submerged in freshwater, water molecules move into the cell.

Raja-Yoga is divided into eight steps, the first is Yama -- non - killing, truthfulness, non - stealing, continence, and non - receiving of any gifts.

Next is Niyama -- cleanliness, contentment, austerity, study, and self - surrender to God.

-----------INPUT END---------------

Generated questions.

Question: What is Flute?

Question: What does Akhil play?

Question: What is Polsambol?

Question: What does Mahagamasekara write?

Question: What is Osmosis?

Question: What is solvent?

Question: What is cell?

Question: What is example?

Question: What is cell?

Question: What is Raja-Yoga?

Question: What is Niyama?

And also, the System can also activate the verbose mode by -v argument to further understand the question generation process. Those results given bellow.

Output: with verbose option.

```
Flute is an Indian classical instrument.

TAGS: [(' Flute ', 'NNP'), ('is', 'VBZ'), ('an', 'DT'), ('Indian', 'JJ'), ('classical', 'JJ'), ('instrument', 'NN')]

{'NN': 5, 'JJ': 3, 'VBZ': 1, 'DT': 2, 'NNP': 0}

 Question: What is Flute?


 --------------------

Akhil plays Flute and Guitar.

TAGS: [('Akhil', 'NNP'), ('plays', 'VBZ'), (' Flute ', 'NNP'), ('and', 'CC'), ('Guitar', 'NNP')]

{'CC': 3, 'VBZ': 1, 'NNP': 0}

 Question: What does Akhil play?

 --------------------


Polsambol is a Sri Lankan dish made of rice and tamarind.

TAGS: [(' Polsambol ', 'NNP'), ('is', 'VBZ'), ('a', 'DT'), ('Sri', 'JJ'), ('Lankan', 'JJ'), ('dish', 'NN'), ('made', 'VBN'), ('of', 'IN'), ('rice', 'NN'), ('and', 'CC'), ('tamarind', 'NN')]

{'JJ': 3, 'IN': 7, 'NNP': 0, 'DT': 2, 'NN': 5, 'CC': 9, 'VBZ': 1, 'VBN': 6}

 Question: What is Polsambol?
```

Mahagamasekara writes books.

TAGS: [(' Mahagamasekara ', 'NNP'), ('writes', 'VBZ'), (' books ', 'NNS')]

{'VBZ': 1, 'NNS': 2, 'NNP': 0}

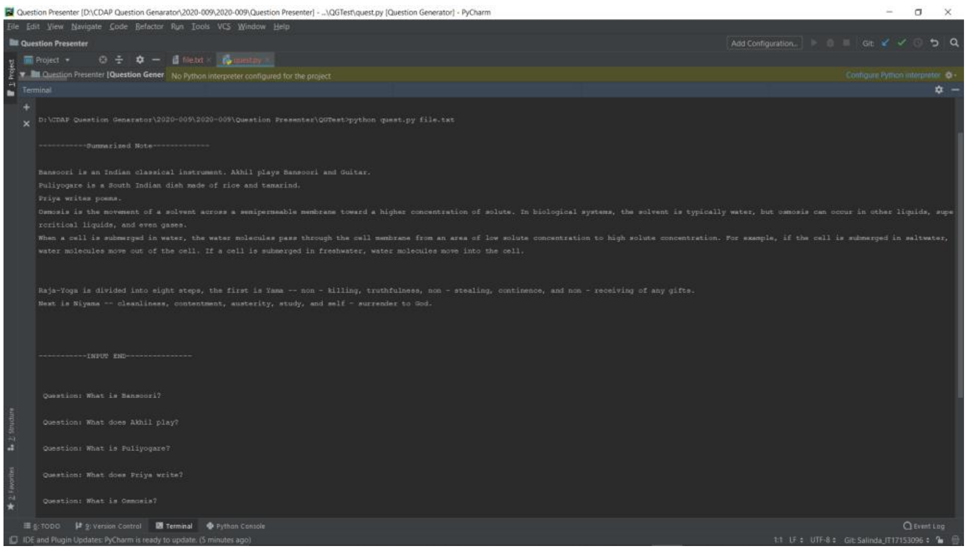
Question: What does Mahagamasekara write?

Osmosis is the movement of a solvent across a semipermeable membrane toward a higher concentration of solute.

TAGS: [('Osmosis', 'NN'), ('is', 'VBZ'), ('the', 'DT'), ('movement', 'NN'), ('of', 'IN'), ('a', 'DT'), ('solvent', 'JJ'), ('across', 'IN'), ('a', 'DT'), ('semipermeable', 'JJ'), ('membrane', 'NN'), ('toward', 'IN'), ('a', 'DT'), ('higher', 'JJR'), ('concentration', 'NN'), ('of', 'IN'), ('solute', 'NN')]

{'JJ': 6, 'IN': 4, 'DT': 2, 'NN': 0, 'VBZ': 1, 'JJR': 13}

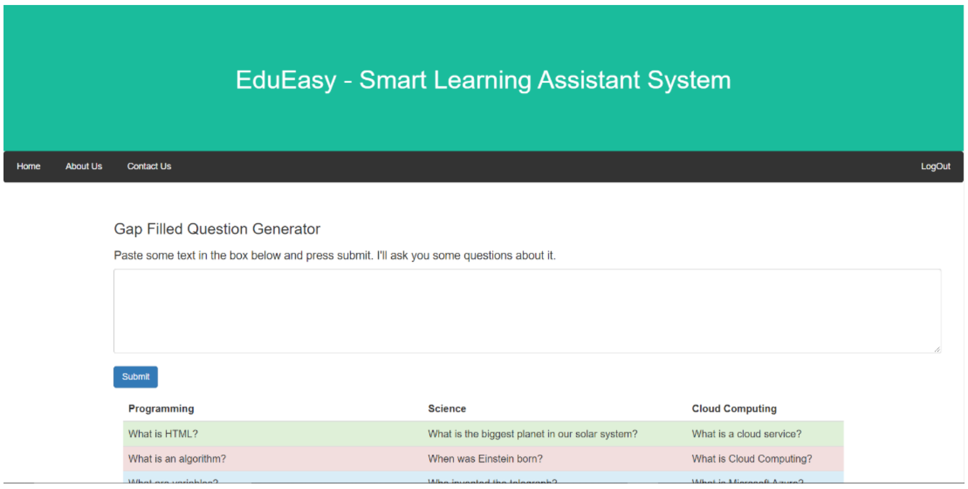
Question: What is Osmosis?


Question Presenter [D:\CDAP Question Genarator\2020-009\2020-009\Question Presenter] - ...\QGTest\quest.py [Question Generator] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Question Presenter
Add Configuration...
Git:
Project
file.txt
quest.py
Question Presenter [Question Gener
No Python interpreter configured for the project
Configure Python interpreter
Terminal
TODO
9: Version Control
Terminal
Python Console
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (5 minutes ago)
1:1 LF UTF-8 Git: Salinda_IT17153096

```
D:\CDAP Question Generator\2020-009\2020-009\Question Presenter\QGTest>python quest.py file.txt

-----------Summarized Note-------------

Bansoori is an Indian classical instrument. Akhil plays Bansoori and Guitar.
Puliyogare is a South Indian dish made of rice and tamarind.
Priya writes poems.
Osmosis is the movement of a solvent across a semipermeable membrane toward a higher concentration of solute. In biological systems, the solvent is typically water, but osmosis can occur in other liquids, supe
rcritical liquids, and even gases.
When a cell is submerged in water, the water molecules pass through the cell membrane from an area of low solute concentration to high solute concentration. For example, if the cell is submerged in saltwater,
water molecules move out of the cell. If a cell is submerged in freshwater, water molecules move into the cell.


Raja-Yoga is divided into eight steps, the first is Yama -- non - killing, truthfulness, non - stealing, continence, and non - receiving of any gifts.
Next is Niyama -- cleanliness, contentment, austerity, study, and self - surrender to God.



-----------INPUT END---------------


 Question: What is Bansoori?

 Question: What does Akhil play?

 Question: What is Puliyogare?

 Question: What does Priya write?

 Question: What is Osmosis?
```

Image - *Question Generator Output*

The next step is to create questions filled-in-the-blank from the given document. The Natural Language Toolkit, or more often NLTK, is a suite of libraries and programs written in the Python programming language for symbolic and mathematical natural language processing for English. This system uses textblob, which is essentially a wrapper around NLTK. Separate the sentences using markers such as full-stop, exclamatory mark, and question mark to trigger fill-in-the-blank questions from the text file. Enable the text file to shape the question as an input to pick a clear, logical sentence from the input document. Blank fill question on chosen descriptive sentences may be asked which is achieved using NLTK. Divide each and every single sentence using a full stop.), (Question mark?) (And Explanatory mark!). Apply the labeling for the POS and get the terms by form. In the case of the word, the sentence introduces the noun, pronoun, adverb, adjective, determiner, superlative degree. Carry out the similar sequence to get the word, pronoun and superlative degree. If the sentence does not contain a subject, a pronoun and a superlative degree than delete the sentence. The result of this diagram is given below.



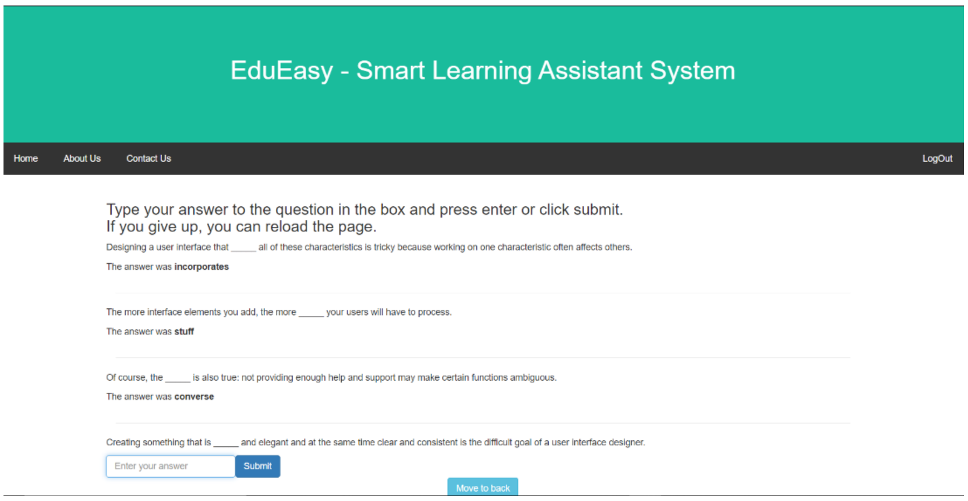
Gap Filled Question Generator

Paste some text in the box below and press submit. I'll ask you some questions about it.

Programming	Science	Cloud Computing
What is HTML?	What is the biggest planet in our solar system?	What is a cloud service?
What is an algorithm?	When was Einstein born?	What is Cloud Computing?
What are variables?	Who invented the telegraph?	What is Microsoft Azure?

Image - ***Gap-Fill-Question Generator Text Input***

After clicking the submit button system generate the gap fill questions and display for the user. Then user can fill those blank space. If user can't find the answer, user can reload the page. Then system automatically display answer for the user. This component results given bellow.



Type your answer to the question in the box and press enter or click submit.
If you give up, you can reload the page.

Designing a user interface that _____ all of these characteristics is tricky because working on one characteristic often affects others.

The answer was **incorporates**

The more interface elements you add, the more _____ your users will have to process.

The answer was **stuff**

Of course, the _____ is also true: not providing enough help and support may make certain functions ambiguous.

The answer was **converse**

Creating something that is _____ and elegant and at the same time clear and consistent is the difficult goal of a user interface designer.

Image - ***Generated gap fill questions.***

The next main goal of this part is to use keywords to check for online answers to similar questions. These outputs will be shown as an interactive search by Google. Question Answering Machine is a machine that addresses questions asked by humans automatically in natural language query. Natural language (e.g. English) is the common way of sharing knowledge. In this program student has to select summarize note from the cloud database which was stored previously. After that student can select the questions which are previously created regarding summarized note. And check synchronism for keywords provided, too. Using machine learning algorithms, this is an advanced Google search. Finally, student can select most suitable answers for given questions. Using this application student can get relevant questions using summarized note. Important thing is student can get correct short answers for this question. This will be to generate questions based on the summarized note (automatic question generation by using NLP) and automatically Google search using question presenter component. This is the block diagram of this system.

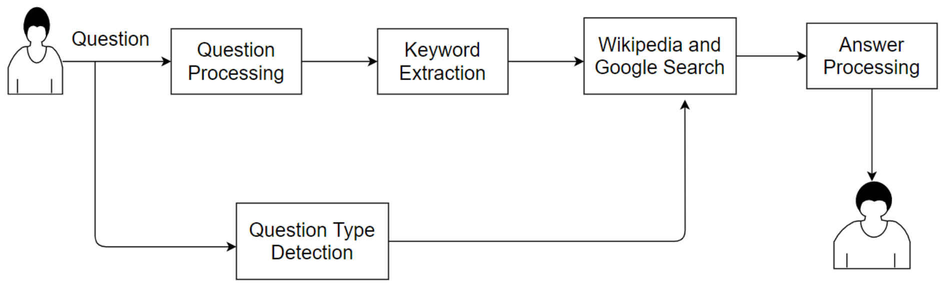
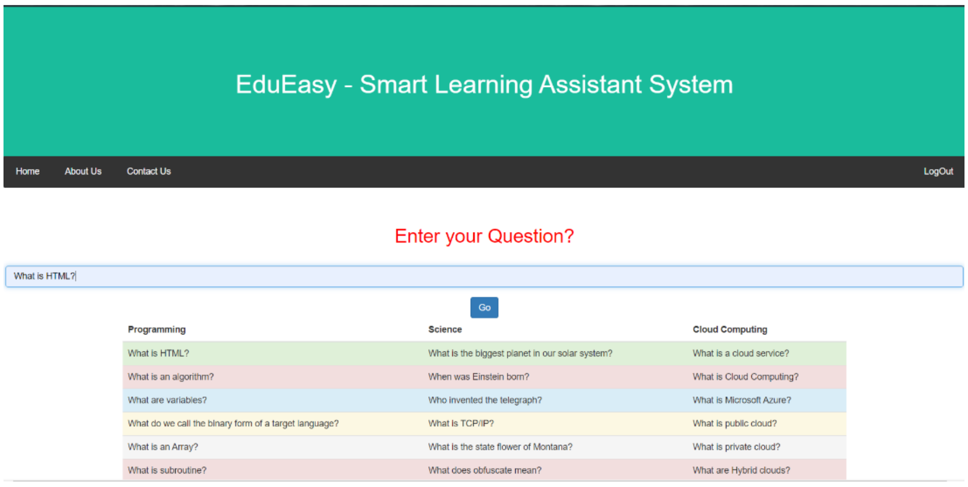


Image - ***Block diagram of the QA System***

Question Answering Systems are mostly separated by two domains. Those are closed and open domain. Closed domain systems answer questions within a specific knowledge domain or to only a certain type of questions. In this system used open-domain. Open-domain systems answer questions about almost everything. These systems rely on far greater volumes of data, using more unstructured data and general ontology, than closed-domain QA systems.

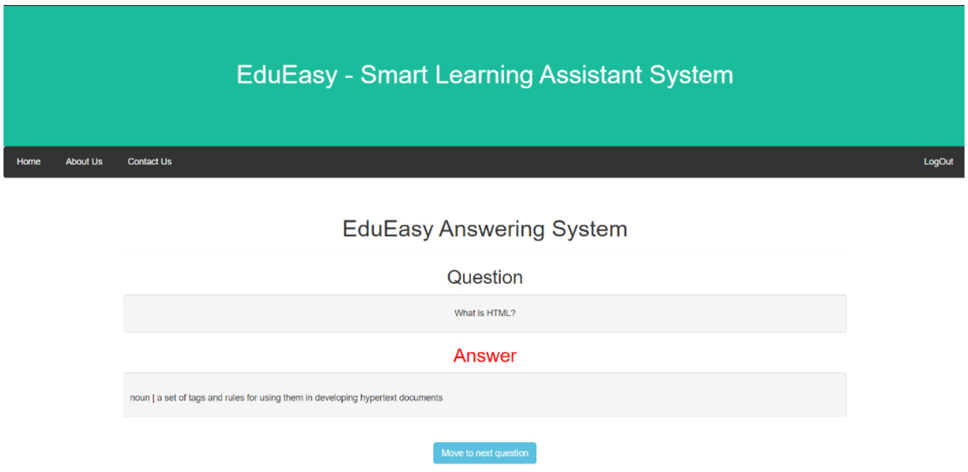
There are lots of search engines available lately. Many of these search engines have tremendous popularity and impressive functionality, but the key issues with these search engines are that instead of providing a clear, correct and reliable response to the user's query they typically have a list of website-related documents that could provide the answer to that question. Although there is a lot of information about the search subject in the list of documents that are collected by the search engine, often it does not include the correct information that the user is searching for. In this method users only have to ask the question and the method will get the most fitting and right answer to that question and send it to the client.

The key goal of the Question Answer method is to provide a concise answer to a user's query instead of digging for a search-related database list. Users only have to ask the question and the device will find the most fitting and right answer for that question and give it to the client. Those answers questions focused on open-domain truth. The Wikipedia and Google search pages are used here to obtain succinct replies.



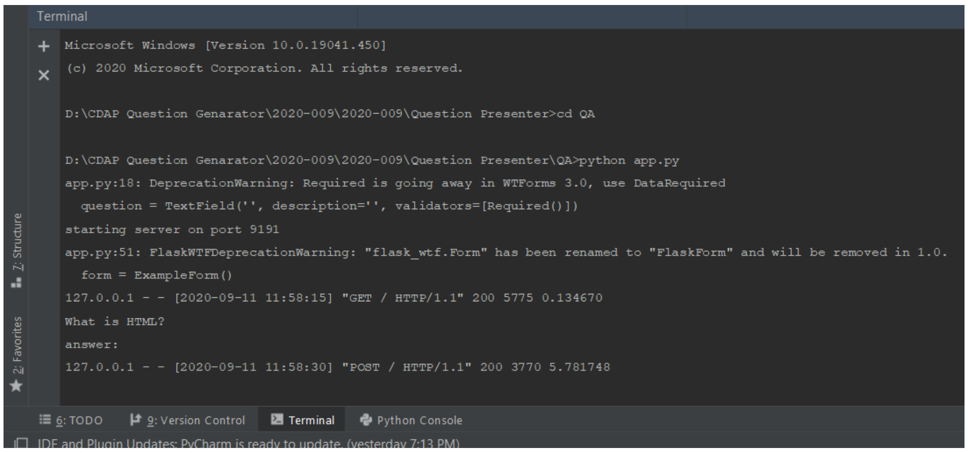
Programming	Science	Cloud Computing
What is HTML?	What is the biggest planet in our solar system?	What is a cloud service?
What is an algorithm?	When was Einstein born?	What is Cloud Computing?
What are variables?	Who invented the telegraph?	What is Microsoft Azure?
What do we call the binary form of a target language?	What is TCP/IP?	What is public cloud?
What is an Array?	What is the state flower of Montana?	What is private cloud?
What is subroutine?	What does obfuscate mean?	What are Hybrid clouds?

Image - *QA System (Enter your question)*



noun | a set of tags and rules for using them in developing hypertext documents

Image - Output ***Answer for your question***



```
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\CDAP Question Genarator\2020-009\2020-009\Question Presenter>cd QA

D:\CDAP Question Genarator\2020-009\2020-009\Question Presenter\QA>python app.py
app.py:18: DeprecationWarning: Required is going away in WTForms 3.0, use DataRequired
  question = TextField('', description='', validators=[Required()])
starting server on port 9191
app.py:51: FlaskWTFDeprecationWarning: "flask_wtf.Form" has been renamed to "FlaskForm" and will be removed in 1.0.
  form = ExampleForm()
127.0.0.1 - - [2020-09-11 11:58:15] "GET / HTTP/1.1" 200 5775 0.134670
What is HTML?
answer:
127.0.0.1 - - [2020-09-11 11:58:30] "POST / HTTP/1.1" 200 3770 5.781748
```

IDE and Plugin Updates: PyCharm is ready to update. (yesterday 7:13 PM)

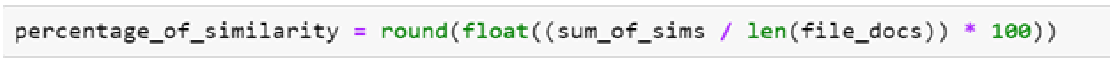
Image - ***QA System running terminal***

Find the most similar title to the summarized note from extracted titles using TFIDF Vectorization.

- Text Extraction

Text analysis is very important before calculating the similarity. The text needs to be extracted for analyzing purpose. sklearn.feature extraction modules are used to extract raw data from a text or an image. The sub module named sklearn.feature extraction.text.TfidfVectorizer converts raw data to a matrix for better visualization. Removing stop words such as 'we', 'is', 'very' is very essential for this process. Those stop words which are represented the text can make the corpus more complex and those uninformative words should be removed. But there can be some texts such as 'I've' I the corpus. after applying the method, the word 'I've' will be tokenized into two tokens ('I' and 've') by default tokenized in CountVectorizer. Then 'I' will be removed since it is a stop word. But 've' will be remained in the corpus which can be given a wrong output. But this Vectorizer has an ability to identify these inconsistences. TfidfVectorizer is a combined of countVectorizer and TfidfTransformer. The function 'fit_transform' as shown in the figure converts tokenized texts into numbers. All the tokens are given numbers and then it will easy for next steps. The 'fit' method computes the weight of every feature. The term 'tf' means term frequency and 'tf-idf' means term frequency times inverse document frequency. Term frequency is counted as the number of times the selected term occurs in the selected document. And idf is calculated using log[(1+n)/(1+df(t)]+1 formula where the n is taken as the total number of documents in the selected set and df(t) is taken as number of documents in the selected set which has the term t. the normalization is also built in the tf-idf vectorization package as an function. As mentioned in above figure 3.5 generated tfidf matrix is multiplied by it's transpose. Sparse matrix is converted into a toarray.

Calculate the percentage of similarity between summarized note and selected lecture using Gensim and TF-IDF models is the next task. Two documents are taken to calculate the percentage of similarity. One document is summarized note and other one is selected lecture note. Before calculating the similarity, all the data should be re-processed. Tokenizing is a major function of Natural language tool kit (NLTK) in Natural Language processing. Figure 3.8 is shown how sentence tokenization is done. It tokenizes each and every sentence in the document. Then as shown in figure 3.9 the words in tokenized sentences are again tokenized using word_tokenize function and converted all letters into lower case letters to improve the accuracy of the results.



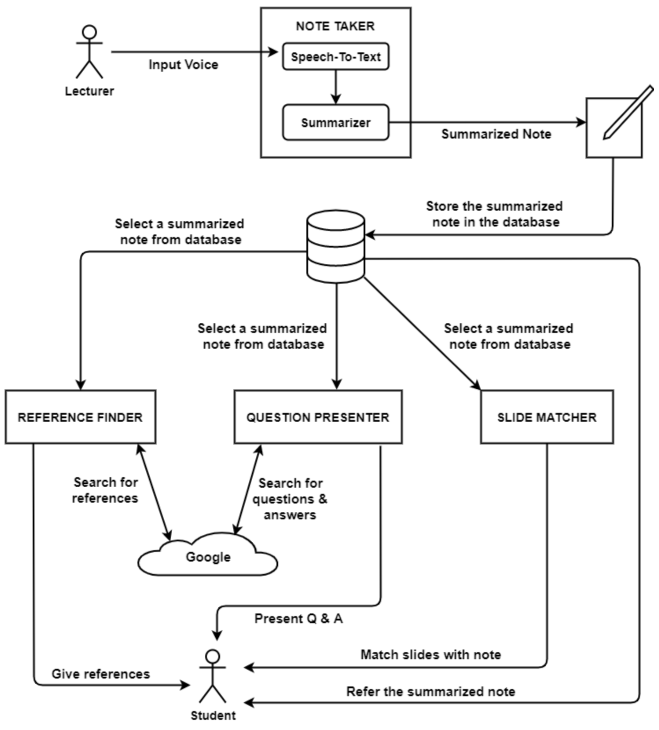
```
percentage_of_similarity = round(float((sum_of_sims / len(file_docs)) * 100))
```

Figure 3.1 – calculating the percentage of similarity

figure 3.1 shown the formula which is used to calculate the percentage of similarity of selected document and summarized note. Length of the documents are specially mentioned because length can be changed from document to document. But the formula was implemented by dividing sum of the similarity of document from the length of the document which can be caused to the final result. Sum of the similarity is calculated using Numpy.

4.1.1. System Architecture

Students who study smart instead of study hard, can have good success in their education. But studying smartly is a challenge without the correct support. EduEasy is an E-learning application for students to effectively learn and revise lectures done at university. EduEasy application has mainly four components. They are note taker, reference finder, question presenter and slide matcher. Each of those components will help to overcome problems which university students has to face on a daily basis. The Figure 2 demonstrates an overview of the EduEasy system.



Lecturer
Input Voice
NOTE TAKER
Speech-To-Text
Summarizer
Summarized Note
Store the summarized note in the database
Select a summarized note from database
Select a summarized note from database
Select a summarized note from database
REFERENCE FINDER
QUESTION PRESENTER
SLIDE MATCHER
Search for references
Search for questions & answers
Google
Present Q & A
Match slides with note
Give references
Refer the summarized note
Student

Figure 2: System Diagram

4.1.2. Project Management

Project management is a fundamental concept that can be put into effect with every project undertaken. This concept is more important to broader tasks carried out by teams and can contribute to project performance by adding project management guidelines to a project. When designing the web application "EduEasy – Smart Learning Assistant System," some of the concepts of project management were implemented in this project in order to achieve a consistent direction that could be traced from the beginning of the project to the end so that the created application fulfills the predefined goals and objectives. Software Development Life Cycle (SDLC) is a structured software construction mechanism that guarantees the consistency and correctness of the developed apps. The goal of the SDLC process is to deliver high quality software that meets the customers' expectations. In the present time frame and cost the system implementation should be complete. SDLC consists of a comprehensive strategy that describes how to develop, create, and manage complex applications. This phase of the SDLC's life cycle has its own mechanism and deliverables that feed into the next phase. SDLC stands for software development.

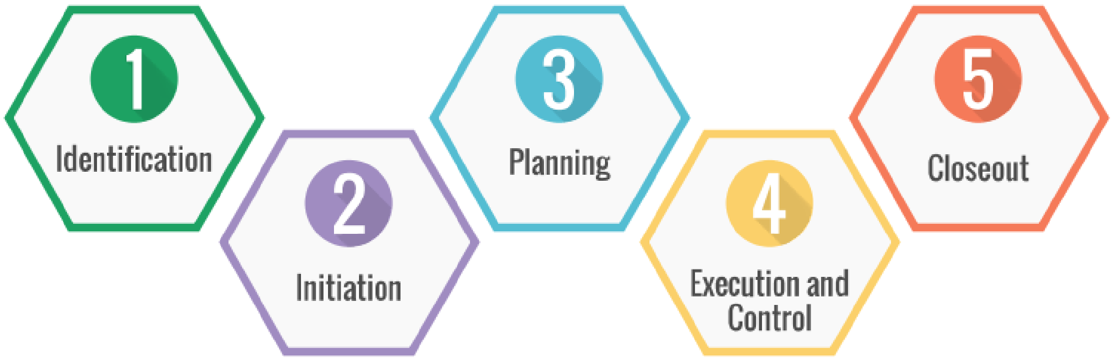


Figure 6: Project Management

Identification and Initiation - The complexity of the project was closely analyzed during the project implementation process in order to identify whether the project could be reasonably completed within the given time period. In dealing with the exact project scale, time constraints became a big concern. Depending on the utility to the customers and depending on the time limit, the features of the programs had to be chosen. In addition, before beginning to prepare the scheme, the feasibility aspect was also considered at this initial stage. In order to determine the best practices available from the current implementations, numerous research have been performed.

Definition and Planning - All the required tasks had to be arranged according to a clear timeline during the project planning period to meet the requirements of the project within the specified timeframe. Even the necessary tools for the project were analyzed during the planning period. Since the project was designed to be carried out as a web application, before beginning the production process, the necessary technology and languages had to be researched and taught. The most suitable tools and software were chosen after review of different tools, considering the appropriateness of such tools and software to the project.

After reviewing many projects and strategies that were pursued within them, it was found that most of the projects that were performed in compliance with a 10 predefined schedule and a time frame were able to effectively complete the project without any difficulties Getting a clear plan often decreased project costs when evaluating commercial projects that are performed by multiple teams. The functional requirements of the project were to be based mainly on the specified time period for implementation. All the roles that had to be created were split down into subtasks with set deadlines for this reason.

Development and Launch - During this deployment process, the planned functions and the pre-considered technology were combined to arrive at a final deliverable. The "EduEasy-Smart Learning Assistant Software" was difficult to create with time limits, and due to the implementation of new technologies not seen in previous programs.
For the implementation scenario an approach such as waterfall or iterative waterfall was not feasible. Therefore, agile approach considering multiple variables was adopted. Agile method has become more practical to use than waterfall or iterative method, because checking can be done by using agile method after each subtask of the software has been developed. With the production process, training can be carried out that helps the team to be versatile during training in modifying multiple parts, making it easier for the team. Using agile approach also helps the team to look ahead of the other activities

to be developed and planned for the implementation process while planning is being carried out and software is being carried out for the project.

In developing the "EduEasy-Smart Learning Assistant System", test driven development was the best approach to be used due the flexibility it provided while developing to meet the exact requirement within the given time constraint. As one subtask could be fully designed, evaluated, iterated until the testing passed and then transferred to the next subtask, the implementation process was more feasible.

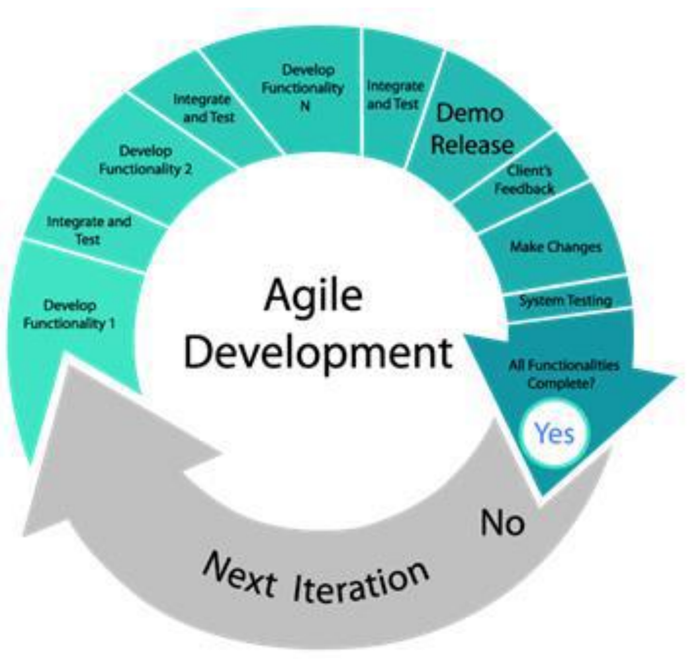


Figure 7: Agile Development

Closeout - This process is largely about launching the final deliverable. The performance of the method may be defined in this stage according to the input provided by the application's users. In order to assess the lessons learnt from the production of the application and the possible improvements that are to be implemented in order to increase the product quality, this stage is so important.

4.1.3. Development

As the backend development programming language of the application, Python has been used as it is more productive language and an interpretive language which is accompanied by elegant syntaxes and libraries. Python is an excellent language choice for scripting and rapid application development in many areas. It is a dynamically typed programming language. For this specific development process, Python version 3.5 was used.

Figure 8: Python Programming Language

Choosing the necessary tools for system development plays a significant role for the success of a program or an application. As this application was designed as a web application and Python 3.5 was used as the programming language, it was decided to use Anaconda Spyder IDE as the development IDE. Spyder IDE is a powerful scientific environment written in Python, for Python. If features unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development too with beautiful visualization capabilities.

Figure 9: Anaconda Spyder IDE

To develop the front-end of the component and connect it to the back-end, Flask framework was used. Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Figure 10: Flask Framework

4.1.4. Software System Attributes

Usability - This would be a strong sign of incorrect goals. The first goal of usability is performance and efficacy, while esthetic meaning comes after the commodity has been proved to be functional. Usability is human-centered, capturing and knowing the entire range of consumer desires is easier than only a collection of diverse data. End-users are the greatest source of details, but the more nuanced products and websites have many user classes, so it is much more important to get all the details, around the board, not just desires but wants and expectations, the complete range of details allows to maintain continuity and authenticity before finalizing the course of usability. Usability innovation can be used to define the needs of consumers for customer retention and performance. In order to meet the needs of the targeted clients according to their criteria, measures must be taken in order to address the needs of the clients and to define their key objectives.

Reliability - In order for an application to be accurate, it should always run without any faults or errors in the stated environment for a given time span. Tech Stability implies operational efficiency. It is defined as the ability of a device or part to perform its necessary functions under static conditions for a specified period of time. Software

Reliability is a core component of software quality, consisting of features, accessibility, consistency, serviceability, ability to update, maintainability, and documentation.

Scalability - Software scalability is the feature of a method or device to expand its capability and flexibility depending on the demand of its users. Scalable applications will stay robust when adapting to updates, improvements, modifications, and resource reduction. In a fast-moving market, you need to develop your products with development in mind. Software scalability is important for the survival of your enterprise. Read on to see if you can make the apps scalable. This is where the scalability of software comes in. Your commodity should be able to fulfill its purpose from now on and can also be modified on the basis of consumer demand. In short, with development in mind, you need to develop your product.

Maintainability - Technology will still require additional functionality or bug fixes. Maintainable software is easy to expand and repair, which enables the uptake and use of software. We will guide you on the design and production of sustainable applications that can help both you and your users. It would be better practice to provide specific names and descriptions in the forms, which would make it simpler later if any improvements were required. As a method for the success of this application, agile principles have been followed. When beginning a project with a simple architecture, it will be easier to extend later without making it more difficult to lead developers in trouble by adding additional features.

There are a variety of explanations for retaining the program after you have delivered it to the customer:

- Bug fixing - It requires looking for and fixing errors to enable the program to run smoothly
- Capability enhancement - Enhancement of software to include the latest features needed by customers
- Replacement -replace unnecessary functionality to increase addictiveness and efficiency
- Security Issues - Fix security vulnerabilities found in your proprietary code or third party code, particularly open source components.

4.1.5. Operations

"EduEasy Smart Leaning Assistant System" web application and the Note Taker component of the system allows users to do following operations.

- User can register to the application.
- User can log in to the application using credentials.
- Users can click "Stat Recording" button in the Speech-to-Text converter and record audio.
- Users can save automatically converted text notes.
- Users can listen to saved notes.
- Users can generate a summarized version of the saved notes.
- Users can generate summaries using any text or by providing a link of a note anywhere from the internet.

4.1.6. User Characteristics

"EduEasy Smart Learning Assistant System" web application mainly targets for university students as well as university lecturers. They are the most suitable group of people to use and experience the benefits of the application.

4.2. Commercialization Aspects of the Product

EduEasy E-learning web application is mainly designed for university environment. Students who wants to study smart, EduEasy system will help them to make their works easy. Students are struggling near exam to find a proper lecture notes, and references, sample questions and the course materials such as lecture PPTs. EduEasy application is a combination of those all requirements.

EduEasy application is introduced with mainly three packages: Platinum, Gold and Silver. All the functionalities are included to the platinum package. Gold package includes all functions except reference finding function. Silver package has only the lecture summarization function. Furthermore, a one-month free trial will be given to all new users of EduEasy system.

Features provided by the EduEasy system

- A summarized note of the lecture conducted by the lecturer for each day.
- Generate automatically the questions and answers for the summarized note.
- Generate automatically the references for additional reading.
- Navigate to the exact lecture slide from summarized note.

Benefits of the EduEasy system

- Easy to find the summarized note.
- Easy to find similar questions and answers.
- Can find relevant references easily.
- Can navigate to the relevant lecture slide automatically.

5. TESTING & IMPLEMENTATION RESULTS & DISCUSSION

5.1. Testing

Testing is the method of evaluating a system or component(s) in order to decide whether or not it meets the specifications defined. Simply put, testing is conducting a method to find any holes, defects or incomplete specifications as opposed to the actual specifications.

And like this, testers too will begin testing. The cost and time to rework and create error-free software that is delivered to the customer is minimized by an early start to testing. However, testing will start from the Requirements Gathering stage in the Software Development Life Cycle (SDLC) and continue until the software is deployed.

When to stop testing is difficult to decide, since testing is a never-ending process and no one can say that a program is 100% tested. To stop the research process, the following factors must be taken into process.

- Testing Deadlines
- Completion of test case execution
- Completion of functional and code coverage to a certain point
- Bug rate falls below a certain level and no high-priority bugs are identified
- Management decision

During the process of testing, there are various stages. A short overview of these stages is given here. Test thresholds provide various methodologies which can be used during software development. The key software testing levels are

- Functional Testing
- Non-functional Testing

Functional Testing - This is a method of black-box testing that is focused on the program requirements that are to be evaluated. The program is checked by supplying input, and the results that need to adhere to the features for which it was intended are then analyzed. Functional program testing is carried out on a complete, integrated system to assess conformity of the system with the defined specifications. Functional testing is a much-needed testing technique for any application as it ensures that the functionality in the application have been developed in 21 accordance with the functions that have been specified. In the “EduEasy-Smart Learning Assistant System” web application, acceptance testing was carried out in order to test for functionality.

Non-functional Testing - Checking an application from its non-functional attributes is based on this section. Non-functional testing includes evaluating applications based on requirements that are non-functional but essential in nature, such as performance, security, user interface, etc.

To test the functionalities of the text summarizer part of the Note Taker component, following test cases was used.

Test Case 01

“Those Who Are Resilient Stay In The Game Longer

On the mountains of truth, you can never climb in vain: either you will reach a point higher up today, or you will be training your powers so that you will be able to climb higher tomorrow. Friedrich Nietzsche.

Challenges and setbacks are not meant to defeat you, but promote you. However, I realise after many years of defeats, it can crush your spirit and it is easier to give up than risk further setbacks and disappointments. Have you experienced this before? To be honest, I don’t have the answers. I can’t tell you what the right course of action is; only you will know. However, it’s important not to be discouraged by failure when pursuing a goal or a dream, since failure itself means different things to different people. To a person with a Fixed Mindset failure is a blow to their self-esteem, yet to a person with a Growth Mindset, it’s an opportunity to improve and find new ways to overcome their obstacles. Same failure, yet different responses. Who is right and who is wrong? Neither. Each person has a different mindset that decides their outcome. Those who are resilient stay in the game longer and draw on their inner means to succeed. I’ve coached many clients who gave up after many years toiling away at their respective goal or dream. It was at that point their biggest breakthrough came. Perhaps all those years of perseverance finally paid off. It was the 19th Century’s minister Henry Ward Beecher who once said: “One’s best success comes after their greatest disappointments.” No one knows what the future holds, so your only guide is whether you can endure repeated defeats and disappointments and still pursue your dream. Consider the advice from the American academic and psychologist Angela Duckworth who writes in Grit: The Power of Passion and Perseverance: “Many of us, it seems, quit what we start far too early and far too often. Even more than the effort a gritty person puts in on a single day, what matters is that they wake up the next day, and the next, ready to get on that treadmill and keep going. I know one thing for certain: don’t settle for less than what you’re capable of, but strive for something bigger. Some of you reading this might identify with this message because it resonates with you on a deeper level. For others, at the end of their tether the message might be nothing more than a trivial pep talk. What I wish to convey irrespective of where you are in your journey is: NEVER settle for less. If you settle for less, you will receive less than you deserve and convince yourself you are justified to receive it.

I recall a passage my father often used growing up in 1990s: “Don’t tell me your problems unless you’ve spent weeks trying to solve them yourself. That advice has echoed in my mind for decades and became my motivator. Don’t leave it to other people or outside circumstances to motivate you because you will be let down every time. It must come from within you. Gnaw away at your problems until you solve them or find a solution. Problems are not stop signs, they are advising you that more work is required to overcome them. Most times, problems help you gain a skill or develop the resources to succeed later. So embrace your challenges and develop the grit to push past them instead of retreat in resignation. Where are you settling in your life right now? Could you be you playing for bigger stakes than you are? Are you willing to play bigger even if it means repeated failures and setbacks? You should ask yourself these questions to decide whether you're willing to put yourself on the line or settle for less. And that's fine if you're content to receive less, as long as you're not regretful later. If you have not achieved the success you deserve and are considering giving up, will you regret it in a few years or decades from now? Only you can answer that, but you should carve out time to discover your motivation for pursuing your goals. It's a fact, if you don't know what you want you'll get what life hands you and it may not be in your best interest, affirms author Larry Weidel: "Winners know that if you don't figure out what you want, you'll get whatever life hands you." The key is to develop a powerful vision of what you want and hold that image in your mind. Nurture it daily and give it life by taking purposeful action towards it. Vision + desire + dedication + patience + daily action leads to astonishing success. Are you willing to commit to this way of life or jump ship at the first sign of failure? I'm amused when I read questions written by millennials on Quora who ask how they can become rich and famous or the next Elon Musk. Success is a fickle and long game with highs and lows. Similarly, there are no assurances even if you're an overnight sensation, to sustain it for long, particularly if you don't have the mental and emotional means to endure it. This means you must rely on the one true constant in your favour: your personal development. The more you grow, the more you gain in terms of financial resources, status, success. If you leave it to outside conditions to dictate your circumstances, you are rolling the dice on your future. So become intentional on what you want out of life. Commit to it. Nurture your dreams. Focus on your development and if you want to give up, know what's involved before you take the plunge. Because I assure you, someone out there right now is working harder than you, reading more books, sleeping less and sacrificing all they have to realise their dreams and it may contest with yours. Don't leave your dreams to chance."

Table 2.3.1: Test Case 1

Test ID	01
Description	Insert the URL of the summarized note into the given text field.
Steps	1. Open the "Reference Finder" interface 2. Navigate to the database where the summarized notes are stored. 3. Copy the link address of the chosen summarized note. 4. Paste the link address of the summarized note in the given space. 5. Press the "Find References" button.
Expected outcome	Display the reference materials in separate tabs on the search engine.
Actual outcome	Display the reference materials in separate tabs on the search engine.

Table 2.3.2: Test Case 2

Test ID	02
Description	Insert two URLs into the given text fields and generate the similarity value.
Steps	1. Open the “Find Similarity Value” interface 2. Navigate to the database where the summarized notes are stored. 3. Copy the link address of the chosen summarized note. 4. Paste the link address of the summarized note in one of the given space. 5. Copy the link address of another reference material and paste it in the other space. 6. Press the “Find Similarity Value” button.
Expected outcome	Display a similarity value.
Actual outcome	Display a similarity value.

5.2. Results

The Speech-to-Text converter was developed in the first half of the note taker process. Both the options of converting a recorded voice clip to text and converting voice to text real time was considered. During the testing, an observation was, converting a long voice record of a lecture to text takes a lot of time to process using the SpeechRecognition API. But when converting speech-to-text real time, processing time is significantly low because it takes a single word or a phrase at a time instead of trying to covert a whole audio file at once. So it was determined that the real time converting method is more suitable because it is time efficient. In the developed speech-to-text converter, Word Error Rate (WER) of the transcriptions are around 30-35% which is better than the normal rate 40-45% [7]. Then the Text Summarizer was developed and a research survey has been conducted to determine the best abstractive text summarizing algorithm for the text summarization process. As the result of that survey (Fig. 6), the TF-IDF algorithm was selected as the most effective algorithm with the choice of 80% respondents. Connected the above two sub-components & tested the note taker using a lecture recording as input. The output summarized note is nearly ¼ long as the full voice transcription.

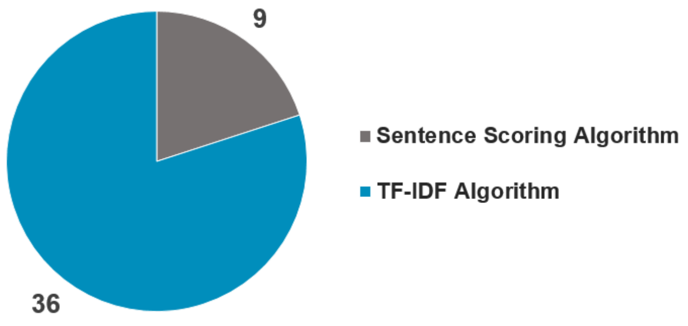


Figure 11: Results of the survey

The text summarizer was developed using the TF-IDF algorithms. Summarized results of the previously stated test cases are shown below.

Test Result 01

“Have you experienced this before? To be honest, I don’t have the answers. Same failure, yet different responses. Who is right and who is wrong? Neither. Each person has a different mindset that decides their outcome. It was at that point their biggest breakthrough came. Perhaps all those years of perseverance finally paid off. It must come from within you. Gnaw away at your problems until you solve them or find a solution. Where are you settling in your life right now? Could you be you playing for bigger stakes than you are? Success is a fickle and long game with highs and lows. Commit to it. Nurture your dreams. Don’t leave your dreams to chance.

Have you experienced this before? To be honest, I don’t have the answers. I can’t tell you what the right course of action is; only you will know. Same failure, yet different responses. Who is right and who is wrong? Neither. Each person has a different mindset that decides their outcome. It was at that point their biggest breakthrough came. Perhaps all those years of perseverance finally paid off. Some of you reading this might identify with this message because it resonates with you on a deeper level. For others, at the end of their tether the message might be nothing more than a trivial pep talk. What I wish to convey irrespective of where you are in your journey is: NEVER settle for less. It must come from within you. Gnaw away at your problems until you solve them or find a solution. Problems are not stop signs, they are advising you that more work is required to overcome them. So embrace your challenges and develop the grit to push past them instead of retreat in resignation. Where are you settling in your life right now? Could you be you playing for bigger stakes than you are? Are you willing to play bigger even if it means repeated failures and setbacks? Are you willing to commit to this way of life or jump ship at the first sign of failure? Success is a fickle and long game with highs and lows. The more you grow, the more you gain in terms of financial resources, status, success. Commit to it. Nurture your dreams. Don't leave your dreams to chance"

There are two main functionalities and two main results in the Reference Finder component. The first part of the reference finding process is started by the extraction of the topic from the summarized note. The reference finder component automatically searches the references according to the keyphrases and demonstrated all the references in separate tabs in the search engine as shown in Figure 3.1.1. The results appear as YouTube videos, articles, books, and blogs, etc.

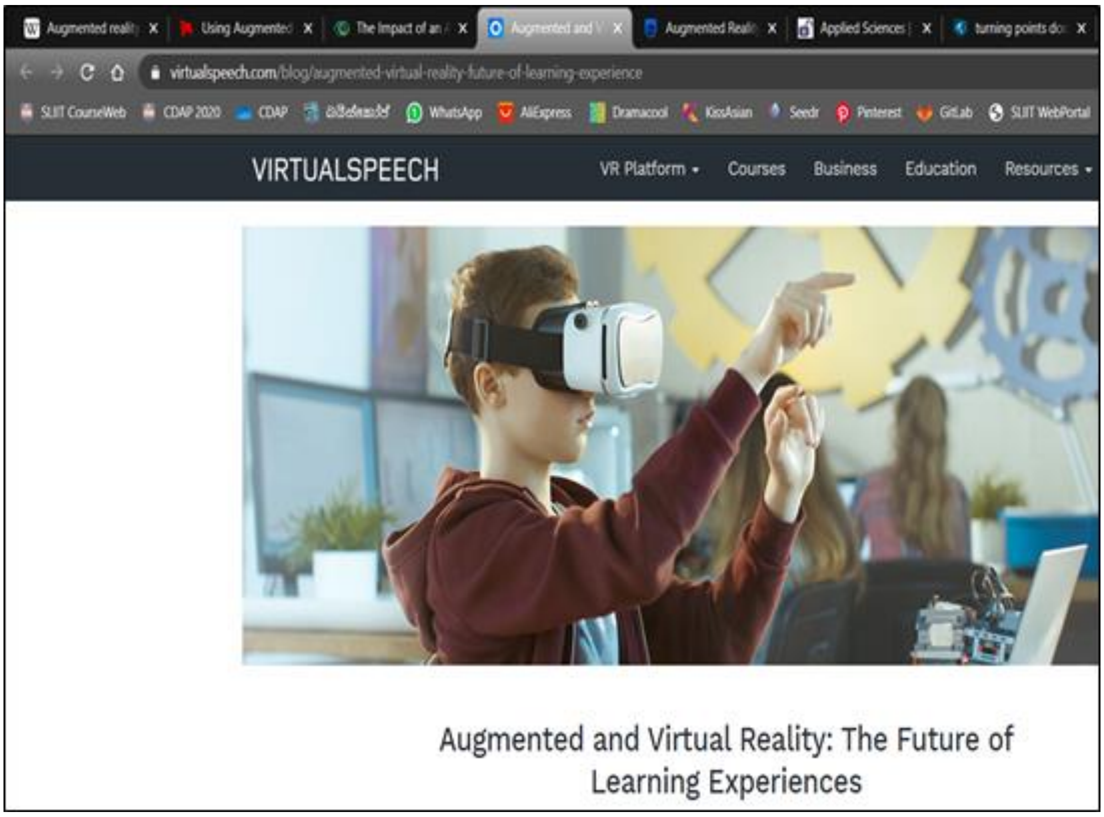
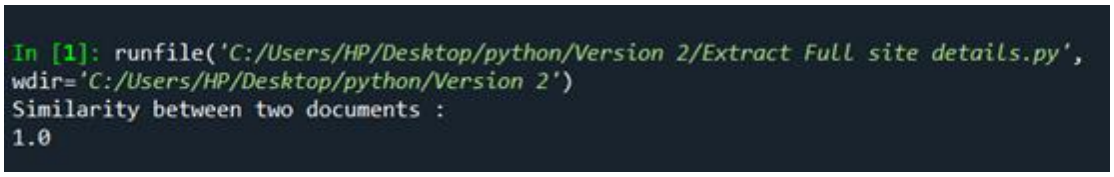


Figure 3.1.1: Automatic reference finder results

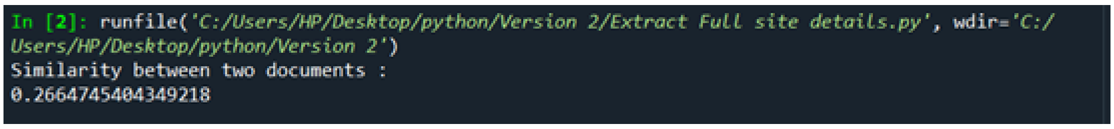
The second part of the component was started by using one of the references that appear as the result in the search engine as the input. The input reference material must not be a video material as a YouTube video. Then the component extracted the content of the web reference and re-processed the data before comparing the content similarity with the summarized note. The similarity algorithm generates a similarity value and the student can get an idea the web results are how much similar to the summarized note.



```
In [1]: runfile('C:/Users/HP/Desktop/python/Version 2/Extract Full site details.py',
wdir='C:/Users/HP/Desktop/python/Version 2')
Similarity between two documents :
1.0
```

Figure 3.1.2: Inputting the same URL as the inputs

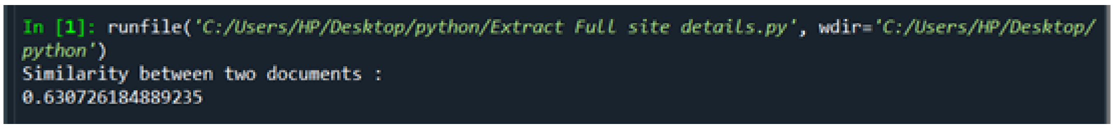
The output that depicts in Figure 3.1.2 is generated by inputting the same document's URL as the two inputs. It generates a 1.0 (1.00%) similarity value.



```
In [2]: runfile('C:/Users/HP/Desktop/python/Version 2/Extract Full site details.py', wdir='C:/
Users/HP/Desktop/python/Version 2')
Similarity between two documents :
0.2664745404349218
```

Figure 3.1.3: Inputting different URLs as the inputs

The output that depicts in Figure 3.1.3 is generated by inputting two different document's URLs as the inputs. It generates a 0.26647 (0.266%) similarity value.



```
In [1]: runfile('C:/Users/HP/Desktop/python/Extract Full site details.py', wdir='C:/Users/HP/Desktop/
python')
Similarity between two documents :
0.630726184889235
```

Figure 3.1.4: Inputting the summarized note's URL and the reference document's URL as the inputs

The output that depicts in Figure 3.1.4 is generated by inputting the summarized note's URL and the reference document's URLs as the inputs. It generates a 0.63072 (0.630%) similarity value.

According to the results, the similarity between the two contents is increased when the similarity value is higher. Finally, the student can refer to the most relevant and rated references using the reference finder component.

5.3. Research Findings

The main outcome of this application is to develop an e-learning application for students to effectively learn and revise lectures done at the university. The Note Taker component is developed as the speech-to-text converter & the text summarizer tool of the application. Following are research findings related to the Note Taker component.

- Converting a long voice record into text takes a lot of time to process than converting speech-to-text real time using the Web Speech API.
- Word Error Rate (WER) is higher when converting speech-to-text real than converting the whole voice record at once.
- 80% of the respondents of the survey selected the summary of TF-IDF algorithm rather than the summary of sentence scoring algorithm, so as conclusion, TF-IDF algorithm is a more effective text summarizing algorithm.
- The Reference Finder application is the main new research finding. Because there is no such application that has all the functionalities similar to the Reference Finder application as one product.
- In addition to that, the automatic reference finder that searches all the types of reference materials such as YouTube videos, pdf, articles, and e-books according to key phrases is one of the research findings. There are no such tools or applications that search all types of reference materials anywhere on the internet. And also it shows user target recommendations in separate tabs in the search engine.
- The other research finding is finding the most relevant reference material by comparing the reference material with the summarized note. The application generated a similarity value and the students can easily measure similarity and refer the suitable reference.

5.4. Discussion

- When the Web Speech API tries to convert a long voice recording into text, it takes a significantly long time to process because it tries to process the whole audio file at once. But when converting speech-to-text real time, API converts words or phrases at a time. So it takes a lower amount of time to process.
- Accuracy of real time speech-to-text conversion is significantly low because the program tries to identify audio wave through a microphone and transcribe at the same time. Even the background noises in the environment can affect the outcome. But when converting an audio file directly, the program does not need to listen through microphones. So from the accuracy standpoint, converting an audio file is better than converting speech-to-text real time.

6. CONCLUSION

The project “EduEasy - Smart Learning Assistant System” is a E-learning web application developed using a combination of several latest technologies which helps university students to overcome some of their daily problems and effectively learn and revise lectures done at the university. The main focus of the application is to help students to make summarize notes, find online references, generate lecture related questions and matching the notes and lecture slides for ease of use. To achieve that goal, applications has four different components which are The Note Taker, The Reference Finder, The Question Presenter and The Slide Matcher.

The application is still in the developing and improving stage. Future goal is to deploy the application for active usage and beta testing with government and private universities with the help of Ministry of Education. Then the application will be released for commercial use with different levels of functionalities.

7. REFERENCES

[1] J. Meer, “Students' note-taking challenges in the twenty-first century: Considerations for teachers and academic staff developers,” in Teaching *in Higher Education, 17,* February, 2012, pp. 13-23.

[2] M. A. Awar, “Pioneering smar learning”. [Online]. Available: https://www.ellucian.com/emea-ap/insights/pioneering-smart-learning. [Accessed May 4, 2020].

[3] A. Jokiaho, B. May, M. Specht and S. Stoyanov, “Barriers to suing E-leaning in an Advanced Way,” in *International Journal of Advanced Corporate Learning,* 2018, vol. 11, no. 1, pp. 17-22.

[4] D. Kaur, “How Smart Class Techology is Benefiting Education Sector”. Available: https://www.entrepreneur.com/article/322587. [Accessed May 4, 2020].

[5] CAE Team, “What is Smart Learning and why does it interest educational centers?” cae.net, 2020. [Online]. Available: https://www.cae.net/what-is-smart-learning-and-why-does-it-interest-educational-centers/. [Accessed: Feb. 21, 2020].

[6] T. Kawahara, H.Nanjo and S. Furui, “Automatic transcription of spontaneous lecture speech,”in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001, ASRU '01., Madonna di Campiglio, Italy,* 2001, pp. 186-189.

[7] C. Munteanu, G. Penn, and R. Baecker, “Web-Based Language Modeling for Automatic Lecture Transcription,” in *8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium,* August, 2007, pp. 2353-2356.

[8] K. S. Thakkar, R. V. Dharaskar, and M. B. Chandak, “Graph-Based Algorithms for Text Summarization,” in *3rd International Conference on Emerging Trends in Engineering and Technology, Goa, India,* 2010, pp. 516-519.

[9] N. K. Nagwani, S. Verma, “A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm,” in *International Journal of Computer Applications,* March, 2011, vol. 17, no. 2.

[10] J. N. Madhuri, R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," in *2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India*, 2019, pp. 1-3.

[11] LajanugenLogeswaran, HonglakLee, DragomirRadev "Sentence Orderingand Coherence Modeling Using Recurrent Neural Networks" Department of Computer Science & Engineering, University of Michigan 2Department of Computer Science, Yale University, The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)

[12] Pantulkar Shravanthi, Dr. B. Srinivasu "Semantic similarity between sentences" M.tech Scholar Dept. of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana- Hyderabad, India, International Research Journal of Engineering and Technology

[13] Adrien Sieg "Text Similarities : Estimate the degree of similarity between two texts". Available (https://medium.com/@adriensieg/text-similarities-da019229c894)

[14] Xiaohao Yang, Jia Liu, "Semi-supervised learning of dialogue acts using sentence similarity based on word embeddings" department of electronic engineering, Tsighua university Beijing100084, China.

[15] Omid Shashmirzadi, Adam Lugowski and Kenneth Younge, "Text Similarity in Vector Space Models. AComparative Study," IEEE International Conference on Machine Learning and Application (ICMLA), Boca Raton, FL, USA, USA, Dec 2019 pp, 19378206

[16] A.Baid, I. Rae, J. Li, A. Doan, and J.Naughton "Toward Scalable Keyword Search over Relational Data," Proceedings of the VLDB En-dowment, vol. 3, no. 1, pp. 140–149, 2010.

[17] Sanglap Sarkar, Venkateshwar Rao, Baala Mithra SM, Subrahmanya VRK Rao(2015)"NLP Algorithm Based Question and Answering System", Proceedings of 2015 Seventh International Conference on Computational Intelligence, Modelling and SimulationPages 97-101

[18] B. B. Dalvi, M. Kshirsagar, and S. Sudarshan "Keyword Search on External Memory Data Graphs," Proceedings of the VLDB Endowment, vol. 1, no. 1, pp. 1189–1204, 2008.

[19] J. X. Yu, L. Qin, and L. Chang, Keyword d Browsing in Databases using BANKS," in Proceedings of the 18th International Conference on Data Engineering ser. ICDE '02, February 2002, pp. 431–440.

[20] J. Coffman and A.C. Weaver, "A Framework for Evaluating Database Keyword Search Strategies," in Proceedings of the 19th ACM Inter-national Conference on Information and Knowledge Management,ser. CIKM '10, October 2010, pp. Search in Databases, 1st ed. Morgan and Claypool Publishers, 2010.

[21] Maria Vargas-Vera and Miltiadis D Lytras. Aqua: A closed-domain question answering system. Information Systems Management, 27(3):217– 225, 2010.

[22] W. Webber, "Evaluating the Effectiveness of Keyword Search," IEEE Data Engineering Bulletin, vol. 33, no. 1, pp. 54–59, 2010

[23] Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. Natural language question generation using syntax and keywords. In Proceedings of QG2010: The Third Workshop on Question Generation, pages 1–10, 2010.

[24] Husam Ali, Yllias Chali, and Sadid A Hasan. Automation of question generation from sentences. In Proceedings of QG2010: The Third Workshop on Question Generation, pages 58–67, 2010. [3] Lorin W Anderson, D

[25] Andrea Varga and Le An Ha. Wlv: A question generation system for the qgstec 2010 task b. In Proceedings of QG2010: The Third Workshop on Question Generation, pages 80–83, 2010.

[26] Aimee A. Callender and Mark A. McDaniel. The bene_ts of embedded question ad-juncts for low and high structure builders. Journal Of Educational Psychology (2007), pages 339{348, 2007.

[27] Arthur C Graesser and Natalie K Person. Question asking during tutoring. American educational research journal, 31(1):104{137, 1994.

[28] P.D. Turney, "Learning Algorithms for Keyphrase Extraction," Information Retrieval, vol. 2, pp. 303-336, 2000.

[29] E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G.NevillManning, “Domain-Specific Keyphrase Extraction,” Proc. 16th Int’l Joint Conf. Artificial Intelligence, 1999.

[30] Manish Agarwal, Rakshit Shah, Prashanth Mannem, Automatic question generation using discourse cues, In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, Association for Computational Linguistics, pp. 1–9, 2011.

[31] X.Somg, J.Feng, G.Li, and Q.Hong, “LCA-based Keyword Search for Effectively Retrieving Information Unit from Web Pages,” Ninth International Conference on Web-Age Information Management, Zhangjiajie Hunan, China, July, 2008.